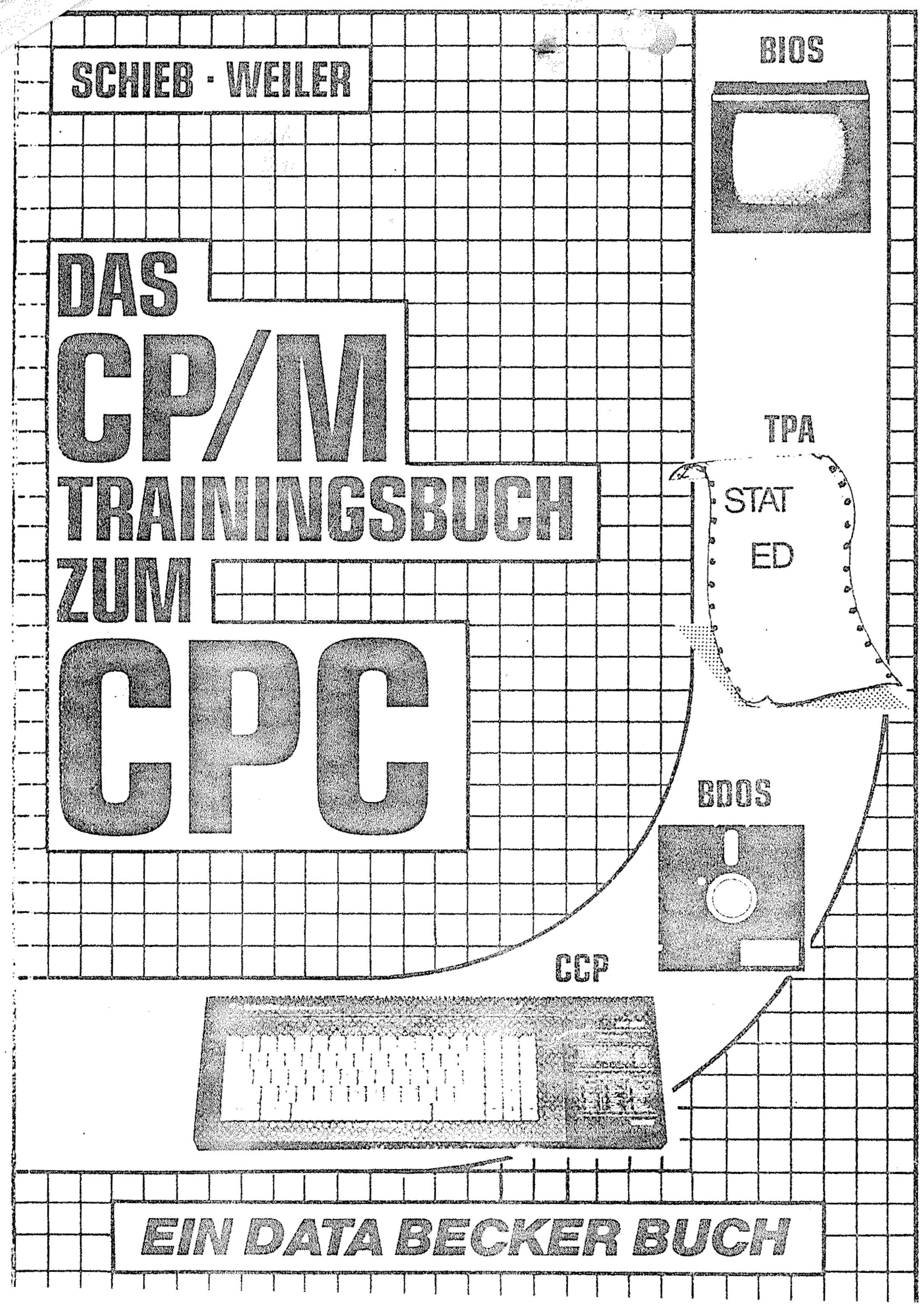

SCHIEB · WEILER
BIOS
DAS
CP/M
TRAININGSBUCH
ZUM
CPC
TPA
STAT
ED
BDOS
CCP
EIN DATA BECKER BUCH


SCHIEB · WEILER
DAS
CP/M
TRAININGSBUCH
ZUM
CPC
BIOS
TPA
STAT
ED
BDOS
CCP
EIN DATA BECKER BUCH


SCHIEB · WEILER
BIOS
DAS
CP/M
TRAININGSBUCH
ZUM
CPC
TPA
STAT
ED
BDOS
CCP
EIN DATA BECKER BUCH

ISBN 3-89011-089-4

Wichtiger Hinweis:
Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technischen Angaben und Programme in diesem Buch wurden von dem Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

INHALTSVERZEICHNIS

I. Der Computer

Um ehrlich zu sein: Ich beneide Sie, weil Sie schon wissen, was Sie nicht wissen. Ich dagegen kann nur versuchen mir vorzustellen, wie weit Ihr Wissensstand in Beziehung auf Computer und alles was dazugehört gediehen ist. Da ich nun leider nicht jeden einzelnen fragen kann, fange ich in diesem Kapitel sozusagen bei "Adam und Eva" an, damit wir alle eine gleiche Ausgangsbasis haben und uns nachher richtig verstehen. Solche Vorgehensweisen gibt's überall, sehen Sie sich nur die DIN-Normen an oder versuchen Sie einmal ein Programm für Ihren Computer zu schreiben. Immer greifen die Menschen auf festgelegte Vereinbarungen zurück, um sich richtig zu verstehen.

Also los. Erst einmal, was ist eigentlich ein Computer? Getreu einer hochwissenschaftlichen Methode wollen wir nun klären, was ein Computer ist.

Dazu stellen wir uns erst einmal ganz dumm und sagen: "Ein Computer ist ein Kasten, in dem etwas passiert". Glücklicherweise weist der englische Name "Computer" schon darauf hin, was da passiert. "Compute" läßt sich mit "rechnen" übersetzen und der Computer ist demnach ein Rechner. Solch einen Zahlenjongleur kann man hervorragend einsetzen, um große Zahlenmengen in der Buchhaltung oder bei der Auswertung von Meßreihen abarbeiten zu lassen. Schüler und Studenten finden sicherlich andere praktische Anwendungen eines elektronischen Rechenknechtes.

I.1 Die Tastatur

Aber - da ist ein Problem. Wir haben nur einen Kasten vor uns der rechnen kann. Wie aber geben wir ihm ein, was er rechnen soll? Eine Möglichkeit wäre, ihm etwas ins Ohr zu flüstern. Aber diese Art der Verständigung mit Computern ist zur Zeit noch Sache der Science Fiction und für uns nicht brauchbar. Deshalb entsinnen wir uns derguten alten Tastatur, und schließen ein solches "Buchstaben- und Zeichen-Eingabe-Gerät" an den Kasten an.

Da der Computer aber sehr viel mehr "drauf" hat als eine Schreibmaschine, muß auch die Tastatur etwas anders aussehen. Zwischen 60 und über 100 Tasten, je nach Preislage und Komfort bietet eine Computer-Tastatur. Häufig gibt's neben der sauberen Ordung von Buchstaben, Ziffern und Zeichen noch eine abgeteilte Zehnertastatur für die schnelle Eingabe von Zahlenkolonnen und, wenn man einen komfortablen Computer hat, auch noch ein Extra-Feld mit sogenannten Funktionstasten.

Ob Ihr Computer eine deutsche oder eine amerikanische Tastatur besitzt, können Sie leicht mit einem Blick feststellen. Steht in der obersten Buchstaben-Reihe von links das Wort QWERTZ, sind Sie glücklicher Besitzer einer Eingabeeinheit nach deutschem Geschmack. Finden Sie dort QWERTY, haben Sie es unter Umständen mit vielen Computer-Programmen leichter, weil die zur Zeit meist aus Amerika kommen, müssen dafür aber auf die hübschen deutschen Buchstaben "ÖÄÜß" verzichten. Man kann eben nicht alles haben.

Ob deutsch oder englisch, es gibt ein paar Tasten, die für die Computerei sehr wichtig sind und oft auf beiden Tastatur-Arten gleich oder ähnlich beschriftet sind. Weil diese Ausnahmen für den Rest des Buches und bei allen

Programmen wichtig sind, erfahren Sie jetzt, um welche Tasten es sich dabei handelt:

WAGENRÜCKLAUF

Sie kennen diese Taste von der Schreibmaschine her, verwenden sie aber beim Computer meist in einem völlig anderen Sinne. Die Abkürzungen auf dieser Taste sehen meist so aus:

RETURN (engl. für "Wagenrücklauf")

ENTER (Abschluß einer Eingabe)

CR ("Carriage return" = engl. Wagenrücklauf)

<--- (erklärt sich von selbst)

RÜCKWÄRTSSCHRITT

BS (kommt von "Back Step")

LÖSCH- ODER KORREKTURTASTE

DELETE (engl. für "löschen")

DEL (Abkürzung für DELETE)

RUB OUT (engl. für "ausradieren")

UMSCHALTUNG

SHIFT (schaltet auf Großbuchstaben um)

FESTSTELLER

```
SHIFT LOCK (schaltet dauerhaft auf Großbuchstaben um)
```

Die genaue Funktion dieser letzten Taste hängt von der Bauart Ihrer Tastatur ab. Teilweise wird mit dieser Taste nicht die gesamte Tastatur umgeschaltet, sondern wirklich nur auf die Großbuchstaben. Das kann eine erhebliche Arbeitserleichterung sein. Auf einigen Tastaturen gibt's für die Umschaltung der Buchstaben noch eigene Tasten die mit

ALPHA LOCK oder

CAPS LOCK

benannt sind. Wenn Sie ganz großes Glück haben, wird der jeweilige Zustand der Tastatur auch noch durch eine Leuchtdiode angezeigt.

Ganz wichtig beim Computern ist die Taste CONTROL, rechts außen auf Ihrer Schneider-Tastatur die für viele Steuerbefehle in Verbindung mit einer Buchstaben oder Zifferntaste gedrückt werden muß. Sie ist in der Regel mit:

CTRL oder (beim Schneider)

CTL

beschriftet.

I.2 Der Bildschirm

Nachdem wir nun die Möglichkeit haben, Daten in den Kasten einzugeben, brauchen wir natürlich auch einen Weg, auf dem die Daten wieder zu uns gelangen können. Es nützt uns ja herzlich wenig, wenn der Rechner die tollsten Berechnungen ausführt, uns aber nicht mitteilen kann, was er herausbekommen hat. Recht umweltfreundlich funktioniert die Datenausgabe über einen Bildschirm, weil es erstens schnell geht und zweitens keinen Lärm macht. Daten und Fehlermeldungen des Computers werden schnell sichtbar und wenn man sich seine Fehler aufschreiben möchte, kann immer noch ein Drucker in Aktion treten.

Ein Bildschirm ist im Grunde ein "umgebauter" Fernseher, der nach dem gleichen Prinzip funktioniert. Nur werden auf dem "Computer-Fernseher", häufig auch Monitor genannt, meist keine laufenden Bilder dargestellt, sondern Buchstaben und Ziffern und sonstige Zeichen. Deshalb sind die Anforderungen an einen Monitor auch andere als an einen Fernseher. Die Auflösung sollte deutlich besser sein, damit man bei den meist stundenlangen Sitzungen keine "Mattscheibe" bekommt und die Bindehaut des Auges nicht über die Maßen strapaziert wird.

I.3 Der Drucker

In der frühen Zeit der Groß-Computer mit heute winzigen Leistungen dienten Fernschreiber als Ausgabemedium für Daten in dauerhafter Form. Der Computer schickt also seine Daten zum Drucker und verbraucht dort in der Regel eine große Menge Papier. Weiterer Nachteil neben dem großen Papier-hunger ist die lautstarke Art, wie Drucker ihre Zeichen zu

Papier bringen. Aber - zur Zeit bestehen noch viele Leute auf Informationen in geschriebener Form und man kommt um einen Drucker nicht herum. Weil es so viele verschiedene gibt, hier ein kurzer Überblick:

Preise und Leistungen schwanken in einer Breite, die selbst von Spezialisten kaum noch zu übersehen ist. So kosten Drucker zwischen 400 und 10 000 Mark und bieten dafür Schreibgeschwindigkeiten zwischen 15 und 800 Zeichen pro Sekunde.

I.3.1 Nadeldrucker

Besonders gut eingeführt als Mitarbeiter von Micros sind die sogenannten Nadeldrucker. Sie haben im Markt die größte Bedeutung und aus ihren Reihen kommen sowohl die preisgünstigsten, wie auch die schnellsten Exemplare. Begonnen hat alles mit Nadel-Druckern, die fein säuberlich Punkt neben Punkt setzten und damit eine Schrift produzierten, die alles andere als säuberlich aussah. Das liegt am Druck-Prinzip der "frühen" Jahre (etwa 1980). Einzeln steuerbare Nadeln drücken auf ein Farbband und dann auf's Papier, so daß hier ein kleiner Punkt abgebildet wird. Aus sieben Punkten läßt sich schon etwa ein Buchstabe oder eine Zahl erkennen. Nur fällt das Lesen längerer Texte recht schwer.

Aber - diese Zeiten sind vorbei. Außer bei ganz billigen Geräten bieten Nadeldrucker heute eine Schreibbreite von 80 Zeichen, etwa 100 Zeichen pro Sekunde an Geschwindigkeit und ein Schriftbild, das zwar nicht für Korrespondenz, aber für nebensächlichen Schriftverkehr vertretbar ist. Großer Vorteil der Nadel-Drucker: Sie bieten in der Regel eine Vielzahl von Schriften und auch frei lad- oder programmierbare Zeichensätze an, die einfach umgeschaltet werden können. So

wird Schriftverkehr mit Griechenland oder auch Japan sehr erleichtert.

Hochleistungs-Drucker erreichen Geschwindigkeiten bis zu 800 Zeichen pro Sekunde bei einer Schreibbreite von 132 Zeichen, kosten allerdings zwischen 8000 und 10 000 Mark. Relativ neu sind die Nadeldrucker der "neuen Art".
Sie besitzen meist 24 Drucknadeln und können damit, bei verringerter Geschwindigkeit, Schreibmaschinen-Qualität erreichen. Übliche Werte sind hier rund 70 Zeichen pro Sekunde für Korrespondenz und 140 - 150 Zeichen im Schnellschreib-Modus.

I.3.2 Typenraddrucker

Zur zweiten Drucker-Kategorie gehören die Typenrad-Drucker. Mit hierzu zählen noch die Typenkorb und Kugelkopf-Drucker. Typenkorb und Kugelkopf-Maschinen basieren meist auf früheren Schreibmaschinen-Konstruktionen und bedürfen in der Regel eines hohen Wartungsaufwandes. Zudem ist die Herstellung recht teuer. Typenrad-Drucker ersetzten den Typenkorb oder den Kugelkopf durch eine runde Scheibe, auf der die verschiedenen Buchstaben und Zeichen federnd angebracht sind. Ein kleiner Hammer schlägt die Type auf's Papier, ein Unterschied zu einer Schreibmaschine ist nicht festzustellen. Wenn also die Korrespondenz "wie gedruckt" aussehen soll, kommt ein Typenrad-Drucker in Frage.

Geschwindigkeit: bei den billigen Modellen meist gerade 20 Zeichen pro Sekunde, bei Spitzenmodellen 80 Zeichen.

I.3.3 Tintenstrahldrucker

Deutlich billiger geben sich die neuartigen Tintenstrahl-Drucker. Wie der Nadeldrucker erstellen sie Buchstaben und Zeichen aus einzelnen Punkten. Dabei findet aber keine mechanische Berührung zwischen Druck-Kopf und Papier statt, sondern es werden winzige Farbtröpfchen auf's Papier geschossen. Die Qualität des Schriftbildes ist mit der von einfachen Nadeldruckern vergleichbar. Besondere Vorteile der Tintenstrahl-Drucker: Sehr geringer Geräuschpegel bei etwa 45 dbA (Dezibel A), trotzdem 150 Zeichen pro Sekunde. Nachteil: Kopien sind nur über mehrmaliges Drucken möglich.

I.3.4 Thermodrucker

Den gleichen Nachteil bieten die sogenannten Thermodrucker. Sie erzeugen die Zeichen durch Wärmebeeinflussung der Papieroberfläche. Selbstverständlich ist hierzu ein besonderes Papier notwendig, das mit der Zeit im Licht vergilbt. Dafür sind die Drucker, die auch im Pünktchen-Verfahren arbeiten, extrem preiswert und für Protokoll-Funktionen durchaus noch zu gebrauchen. In Taschenrechnern unterhalb der 100 Mark-Preisgrenze sind die einfachen Thermodrucker heute schon im Einsatz.

Das Optimum für den schnellen, sauberen Druck per Personal Computer sind die Laser-Drucker. Deren Ausstoß ist von einer professionell gesetzten und gedruckten Arbeit nicht zu unterscheiden.

Ach ja, da fällt mir etwas ein. Unser Kasten, der mittlerweile in der Beschreibung einem Computer schon recht ähnlich ist, sollte natürlich auch einen Schalter haben, um

die Stromzufuhr unterbrechen zu können. In der an sich guten Absicht, versehentliches Ein- oder Auschalten zu erschweren, kamen viele Computerhersteller auf den Dreh, daß ein Einschaltknopf irgendwo versteckt sein muß. Wenn Sie an Ihrem Computer den Schalter noch nicht gefunden haben, suchen Sie an einer Stelle, wo ihn kein Mensch vermuten würde. Er ist bestimmt da.

I.4 Datenspeicher

Wir können mit unserem theoretischen Computer nun schon ziemlich viel machen. Der Computer selber ist da, die Tastatur, ein Bildschirm und einen Drucker haben wir auch schon. Was noch fehlt, ist ein Platz, wo die Daten gespeichert werden können. Einmal innerhalb des Computers und - für längere Zeit - auch außerhalb. Grundsätzlich kann man sagen, ist die Art Daten zu speichern in jedem Medium des Computers gleich. Ob im sogenannten Arbeitsspeicher, das ist der, der jetzt gerade meinen hier geschriebenen Text aufbewahrt, oder auf einer Diskette oder einer Hard-Disk. Wieso und wie der Computer etwas speichern kann, sollen Sie jetzt erfahren.

Weil das ein wenig theoretisch ist, holen Sie sich vielleicht ein Tässchen Kaffee, nehmen einen guten Schluck und lesen dann aufmerksam weiter. Ich warte bis Sie wieder da sind...

I.5 Eins oder Null

Sie werden nicht abstreiten, daß ein Computer ein elektrisches Gerät ist. Wenn dem so ist, dann müssen auch die Informationen innerhalb des Computers in einer elektrischen Form verarbeitet werden. Aber, wie stellen wir das an? Wie sollen wir einem Computer sagen was wir meinen? Dazu ist eine Art von Darstellung notwendig, die der Computer versteht. Es gibt eine Art, die der Computer kennt: Die beiden Zustände: Strom fließt - kein Strom fließt. Nach dem gleichen Prinzip funktioniert auch eine Glühbirne. Bemerkt sie, daß der Strom fließt, fängt sie schleunigst zu brennen an. Fällt ihr auf, daß kein Strom mehr da ist, geht sie einfach aus. Was so eine dumme Glühbirne kann, kann unser Computer schon lange. Seine Erbauer haben ihm in sein Elektronen-Gehirn geschrieben, daß der Zustand "Strom fließt" gleichbedeutend ist mit dem Wert "Eins". So wie für uns die "1" auch einen bestimmten Wert hat.

Fließt kein Strom, bedeutet das für den Compi: "0". Damit kann er jetzt zwei Zustände unterscheiden, was zu erstaunlichen Ergebnissen führt.

I.6 Binär zählen

Für den täglichen Gebrauch ist die binäre Zählweise, also mit eins und null, nicht sonderlich gut geeignet. Bis Sie auf diese Weise Ihr Geburtsdatum hergesagt haben, bekommen Sie schon eine trockene Zunge. Glücklicherweise hat der Computer keine Zunge und so kann auch nichts trocken werden, wenn er wie ein Wahnsinniger mit den beiden Ziffern hantiert.

Sehen wir uns das Arbeiten mit Zahlen einmal grundsätzlich an. Es gibt verschiedene Zahlensysteme: Das binäre, das dezimale, das hexadezimale und noch einige mehr. Das dezimale System beherrschen wir alle aus dem Effeff. Wir können kaum anders denken als in Zehnerwerten. Was machen wir aber dabei? Wir denken uns eine Zahl, zum Beispiel die Null. Soll es mehr sein, nehmen wir die Eins und so weiter. Sind wir bei der "9" angelangt, setzen wir die nächste Zahl aus den zwei niedrigsten wieder zusammen, was eine "10" ergibt.

Genau so können wir auch vorgehen, wenn wir nur zwei Ziffern zur Verfügung haben. Der erste Wert ist die Null. Der nächste die Eins. Dann ist Schluß, wir haben keine Zahlen mehr um weiterzuzählen. Also setzten wir die nächste Zahl aus den beiden zusammen und erhalten für den Wert Zwei die Ziffer 10. Die Drei sieht so aus: 11 und für die Vier brauchen wir schon wieder eine neue Stelle, was 100 ergibt.

Zum Verständnis hier nochmal eine Tabelle der binären Zahlen:

```
Null = 0
Eins = 1
Zwei = 10
Drei = 11
Vier = 100
Fünf = 101
Sechs = 110
Sieben = 111
Acht = 1000
```

und immer lustig so weiter.

Achtet man nur auf die Stellen dieses Zahlensystem, kann man mit einer Stelle zwei Werte darstellen, mit zwei Werten vier Stellen, mit drei Stellen acht, mit vier Stellen 16 und so weiter. Von Stelle zu Stelle ergibt sich immer die doppelte Möglichkeit. In unserem Dezimal-System kennen wir die Wertigkeit der Stellen ja auch. Nur geht's hier immer gleich um Zehner. Eine Stelle kann 10 Werte darstellen, zwei Stellen hundert, drei Stellen tausend...

Die Geschichte mit den zwei Zahlen nennt sich binäres (zweizahliges) Zahlensystem und hat den Leuten, die uns nicht nur den Hamburger, sondern auch die Computer schmackhaft machten so gut gefallen, daß sie sich gleich ein paar Fachausdrücke dafür ausdachten.

Eine Stelle einer Zahl heißt im amerikanischen "digit" und zu binär sagt die Super-Nation "binary". Beides zusammen nennt sich "binary digit" und wurde kurzerhand auf "bit" abgekürzt.

Nun ist aber eine einzige Stelle als Speicherplatz recht wenig und moderne Computer haben eine ganze Menge Speicherplätze. Läßt man es bei einem Bit, werden die Zahlen für Speicherplatzangaben enorm groß. Aus diesem Grunde hat man Einheiten, "Wörter", geschaffen, die mit einem Wort oder einer Zahl eine ganze Reihe von Bits beschreiben. Mit einem Acht-Bit-Wort kann man so 256 Bits beschreiben und mit einem 16-Bit-Wort 65536.

Sehr häufig kommt die Einheit mit 256 Möglichkeiten vor, weshalb diese Einheit den Namen "Byte" trägt. Es ist allerdings ein Gerücht, daß die amerikanische Fachzeitschrift mit

dem Namen Byte aus diesem Grunde nur mit maximal 256 Seiten erscheinen darf. Sie schafft in Wirklichkeit mehr als das Doppelte.

I.7 Werte speichern

Um richtig arbeiten zu können, muß der Computer die Zahlen auch speichern können. Eine Zahl, wie wir gesehen haben, besteht für den Computer aus einem elektrischen Zustand: geladen oder ungeladen. Faßt man acht solcher Speicherplätze zusammen, erhält man ein Acht-Bit-Wort, das insgesamt 256 Werte darstellen kann. "Zufälligerweise" entspricht das genau einem Byte. Weil aber auch ein Byte noch nicht sehr viel ist, gibt's noch die Bezeichnung Kilobyte. Dabei entspricht aber ein Kilo nicht genau 1000 Bit sondern, entsprechend der binären Zählweise 1024 Bits. Für ganz große Einheiten gibt's dann noch das Megabyte.

Mit Hilfe elektronischer Schaltungen ist es nun möglich, die Werte aus einem Byte herauszulesen, weiter zu verarbeiten und auch neue Werte wieder einzulesen. Das ist das ganze Geheimnis des Computers. Sein Vorteil ist, daß er dieses Operationen in sehr kurzen Zeitabständen machen kann und deshalb ein echter und nützlicher Datenverarbeiter ist.

Da jede Speicherstelle wirklich vorhanden sein muß, hat jeder Computerspeicher nur einen begrenzten Raum, in dem Daten abgelegt werden können. Übliche Größen für den Arbeitsspeicher eines Personal Computers mit einer 8-Bit-Zentraleinheit, das ist das eigentliche Hirn, sind 64 Kilobyte oder kurz 64 KB. Die neuen 16-Bit-Computer bieten Speichergrößen zwischen 128 KB und rund acht Megabyte.

Etwas unklar ist bis jetzt aber noch, wie man zum Beispiel

Text in einem Computer speichern kann, wo der doch nur Zahlen hin- und herschickt.

In diesem Punkt sind Computer ganz menschlich. Sie benutzen, genau wie wir, wenn wir eine Fremdsprache übersetzen, ein Wörterbuch. Innerhalb des Computers gibt's eine Tabelle, in der für jeden Buchstaben und all die anderen Zeichen ein Zahlenwert festgelegt ist. Kommt nun ein Buchstabe in den Computer rein, schaut der in der Tabelle nach und findet den richtigen binären Zahlenwert. Den wiederum kann er problemlos verarbeiten. Bei der Ausgabe funktioniert die Sache genau andersherum.

Damit das sinnvoll nicht nur auf einem Computer eines bestimmten Herstellers klappt, muß man sich über die "Übersetzungstabelle" einigen. Leider, aus deutscher Sicht, geschah diese Einigung in Amerika und umfaßt auch nur die dort gebräuchlichen Zeichen. Das Ganze nennt sich ASCII (American Standard Code for Information Interchange = amerikanischer Standardcode für Informations-Austausch) und beinhaltet eben die deutschen Sonderzeichen nicht. In diesem ASCII-Code sind 128 Zeichen erfaßt, von denen jedes seinen eigenen Wert hat. Im Anhang finden Sie eine Tabelle mit all diesen Werten.

Was aber macht der Computer mit einem Speicher voller Werte? Nichts. Er ist zu blöde, irgendetwas mit diesen Daten anzufangen. Damit etwas passiert, müssen Sie ihm erst einen entsprechenden Befehl geben. Sie können ihm befehlen, den Text in seinem Speicher mit einem glatten rechten Rand zu versehen oder die Beleuchtung anzuschalten. Erst wenn sie sinnvolle Befehle geben, kann der Computer auch etwas ausrichten. Auf der untersten Ebene sieht das so aus: Sie sagen dem Computer: Nimm das Bit im Speicher "a", addiere es zu dem Bit in Speicher "b" und lege das Ergebnis in Speicher "c" ab.

Eine ganze Reihe von solchen Anweisungen sind bereits in der Central Processing Unit (CPU), dem Rechengehirn des Computers, gespeichert. Aus vielen dieser Operationen lassen sich auch komplizierte Vorgänge zusammenstellen, was die Aufgabe der Maschinen- oder Assembler-Programmierung ist. Das sind Programmier-Sprachen, welche die Maschinenfunktionen direkt steuern. Basic, Pascal oder Fortran steuern die CPU erst nach einer internen Übersetzung des jeweiligen Programms in die Maschinensprache.

Um eine bestimmte Aufgabe zu lösen, muß dem Computer jeder einzelne Schritt haarklein vorgegeben werden. Das geschieht in der Regel mit Hilfe eines Programms, was nichts anders ist, als die Aneinanderreihung von Befehlen, die nacheinander abgearbeitet werden. Das Programm selber muß natürlich auch im Speicher vorhanden sein, damit es ausgeführt werden kann, verbraucht also auch Speicherplatz.

Je nach Anforderungen, kann solch ein Programm zwischen zehn und 120 KByte verbrauchen. Unter Umständen ist der Platz im Speicher allein schon durch das Programm so belegt, daß für die Ausführung des Progamms und die Daten des Benutzers kein Platz mehr bleibt. Das bedeutet dann: Dieses Programm ist für diesen Computer zu groß und kann nicht benutzt werden. Aber diese Fälle sind recht selten. Häufiger kommt es vor, daß die Daten des Benutzers nicht mehr in den eingebauten Arbeitsspeicher passen.

I.8 Massenspeicher

In den seltensten Fällen ist es nötig, ein ganzes Programm auf einmal im Speicher zu haben. Meist genügt es völlig, wenn die wichtigen Teile im Speicher stehen und der Rest nur bei Bedarf nachgeladen wird. Aus diesem Grunde gibt es bei Computern außer dem internen Arbeitsspeicher auch externe Speichermöglichkeiten, die nicht einer so großen Platzbeschränkung unterliegen und außerdem billiger sind. Die sogenannten Massenspeicher können sein: Ein Kassettenlaufwerk, eine Floppy-Disk-Station oder eine Festplatte. Alle drei haben ihre Vor- und Nachteile, bieten aber alle reichlich Platz für Daten und können diese in erträglicher Zeit mit dem Computer austauschen.

I.8.1 Floppy-Disk

Die Floppy-Disk, auch Floppy oder nur Disk genannt, ist eine Art Schallplatte aus dem Material eines Tonbandes. Ein dünne Scheibe mit magnetisierbarer Oberfläche, eingehüllt in einen Schutzumschlag, gegen die fettigen Finger des Benutzers oder auch Staub- und Schmutzteilchen. Auf der Oberfläche der Floppy, wenn sie im Computer steckt, fährt ein Schreib-Lesekopf entlang und macht verschiedene Stellen magnetisch oder nicht.

Da haben wir wieder die zwei Zustände: Magnetisch oder nicht. Das deckt sich mit der Speichermethode im Computer - und das mit Absicht. Die Daten im Computer, die dort als elektrische Zustände vorliegen, lassen sich auf diese Weise auf die Floppy übertragen. Hat ein Bit den Zustand geladen, macht der Schreibkopf der Floppy einen kleinen Punkt der Diskettenoberfläche durch einen Stromstoß magnetisch. Hat

das nächste Bit keine Ladung, kommt auch nichts auf die Floppy. Aus der Folge von magnetischen und nicht magnetischen Punkten auf der Floppy können nun ein Programm oder eine Reihe von Daten gelesen und wieder in den Computer übertragen werden. Damit haben wir die Möglichkeit, Daten extern zu lagern.

Die Diskette dreht sich mit 200 bis 300 Umdrehungen pro Minute und würde ein fürchterliches Datenchaos hinterlassen, wenn man die Daten einfach planlos draufschriebe. Also unterteilt man die Diskette in ein bestimmte Anzahl von Spuren, die nebeneinander auf der Diskette liegen wie die Fahrbahnen einer mehrspurigen Autobahn.

Will man Daten ablegen, fährt der Schreibkopf über die entsprechende Spur und schreibt munter drauf los. Nun ist aber eine solche Spur immer noch recht lang und man müßte ganz schön lange warten, bis man seine Daten von solch einer Spur wieder in den Computer gelesen hat.

Um diese Zeit zu verkürzen, unterteilt man die Diskette nochmal in Sektoren (je nach Hersteller zwischen 128 Bytes und 1024 Bytes groß, der Schneider CPC verwendet 512 Bytes), was dann aussieht wie die Stückchen einer Torte.

Diese Methode der Aufzeichnung ist recht praktisch, verhindert aber unter anderem, daß man einfach eine Diskette des einen Rechners in das Laufwerk eines anderen Rechners reinstecken und die Daten lesen kann. Jeder Hersteller denkt sich nämlich da so seine eigene Methode aus, um uns Anwender zu ärgern. Große Spezialisten auf diesem Gebiet sind die Firmen Apple und Victor.

Wenn Sie neue, unformatierte Disketten kaufen, müssen Sie diese in der Regel erst einmal formatieren. Dazu gibt's bei jedem Computer ein spezielles Programm, das genau die Spuren

und Sektoren auf die Diskette schreibt, die der Rechner braucht.

Wenn Sie sich gerne etwas ärgern wollen, formatieren Sie doch mal die Diskette mit Ihren wichtigsten Daten drauf. Sie werden staunen, wie wenig davon noch übrig bleibt. Deshalb geben Sie Acht und machen Sie einen Schreibschutz auf die Kerbe der Diskette. Die kleinen Aufkleber, die jeder Packung Disketten beiliegen, können Ihnen Kopf und Kragen retten. Bei den 3-Zoll-Disketten wie der Schneider-Computer sie verwendet, verstellt man einen Plastikschieber, was es dem Computer dann unmöglich macht, auf diese Diskette zu schreiben.

Genauso klug ist es übrigens, immer von allen wichtigen Daten eine Sicherheits-Kopie anzulegen und unerreichbar für Kinder, Hunde, Katzen, Feuer, Einbrecher, Schneestürmen, Erdbeben und sonstigem Ungemach aufzubewahren. Wenn Sie schon Ihre Diamanten nicht in den Safe legen - tun Sie's wenigstens mit Ihren Sicherheits-Kopien. Wie Sie mit Hilfe von CP/M Sicherheits- und andere Kopien machen können, erfahren Sie in einem folgenden Kapitel.

Zur Zeit befinden sich ganz verschiedene Disketten-Formate auf dem Markt. Teilweise findet man noch die altehrwürdige 8 Zoll-Diskette, die einmal den Siegeszug der elektronischen Datenverarbeitung einläutete. Besonders weit verbreitet sind die 5 1/4 Zoll-Disketten, die normalerweise in Personal Computern verwendet werden. Das Format ist einigermaßen handlich und mittlerweile passen auch schon fast zwei Megabyte auf solch ein Scheibchen.

Besonders aus Japan dringen aber auch 3 Zoll und 3 1/2 Zoll-Disketten auf den Markt, die ein nochmal verbessertes Handling versprechen, weil der Schreib/Leseschlitz durch eine Metall-Lasche verschlossen ist, solange die Diskette nicht

im Laufwerk steckt. Außerdem haben die 3er Disketten eine feste Plastik-Hülle, so daß man sie nur noch mit Gewalt knicken kann.

I.8.2 Hard-Disk

In Computern, die hauptsächlich beruflich genutzt werden, findet man auch noch die Hard-Disk oder den Festplatten-Speicher. Diese Dinger sind in den Abmessungen einer 5 1/4 Zoll-Diskette gleich und nehmen auch den gleichen Raum in einem Computer-Gehäuse ein (Weil die Verkleinerungswelle vor nichts haltmacht, gibt's natürlich auch schon 3 1/2 Zoll-Festplattenlaufwerke). Innendrin aber drehen sich beschichtete Metallplatten mit gut 3000 Umdrehungen pro Minute und die Schreib/Leseköpfe fliegen im Tiefstflug über die Plattenoberfläche. Ein Staubkorn oder Rauchpartikelchen auf dieser Oberfläche hätte die gleiche Wirkung wie das Matterhorn für einen Jumbo, wenn er zu tief fliegt.

Deshalb kommen die Hard-Disks in einem luftdicht verschlossenen Gehäuse, und darin sollte man sie auch lassen. Als Standard-Fassungsvermögen gelten seit Beginn der 80er Jahre 10 MByte für eine Festplatte. Aber, die Preise sinken und die 20 MB-Platte hält langsam ihren Einzug in die Computer-Gehäuse.

Außer dem besonders hohen Fassungsvermögen bietet die Festplatte aber noch einen entscheidenen Vorteil: Sie ist bis zu 20 mal schneller im Datenzugriff, als eine Diskette.

So, wenn ich richtig mitgezählt habe, müßten jetzt alle Teile eines richtigen Computers besprochen sein und Sie sollten die grundlegenden Kenntnisse über dieses hilfreiche und manchmal auch spaßige Gerät intus haben. Im nächsten

Kapitel erfahren Sie etwas über das Betriebssystem und wofür man sowas überhaupt braucht.

I.9 Was Sie jetzt schon wissen

* *Sie kennen jetzt die Bestandteile eines Computers, von der Tastatur bis zum Bildschirm*

* *Sie kennen die verschiedenen Druckertypen, ihre Vor- und Nachteile*

* *Sie wissen auch was ein Speichermedium ist*

* *Sie kennen die Zahlen Eins und Null, können binär zählen und wissen, wie die Werte gespeichert werden*

II. Das Betriebssystem

Im vorhergehenden Kapitel haben Sie einen kurzen Einblick in die Hardware eines Computers, die einzelnen Bauteile, erhalten. Wenn Sie den Materialwert eines solchen Computers nehmen, ist das Ding nicht viel wert. Grund: Sie können damit praktisch nichts anfangen. Richtig wertvoll wird ein Computer erst, wenn Sie darauf für Sie wichtige und nützliche Programme laufen lassen können. Damit das klappt, brauchen Sie die sogenannte Software. Auch hier müssen wir aber wieder unter verschiedenen Begriffen unterscheiden.

Um die allgemeine Verwirrung klein zu halten, beschäftigen wir uns mit zwei Typen von Software: Dem Betriebssystem und den Programmen. Sie sollten im Hinterkopf behalten, daß natürlich auch das Betriebssystem ein Programm ist. Der Unterschied den ich meine, kommt von der Funktion der unterscheidlichen Software her und ich will Ihnen das gerne erklären.

Häufig höre ich unerfahrene Leute mit Interesse an einem Computer fragen:" Kann der auch Textverarbeitung?" oder so ähnlich. Diese Leute haben den Unterschied zwischen einem Programm und einem Betriebssystem noch nicht verstanden, weshalb hier eine Erklärung folgt, falls Sie auch schon einmal so gefragt haben.

Stellen Sie sich bitte vor, wir haben es nicht mit einem Computer zu tun, sondern mit dem ganz normalen Leben. In einem solchen ist ein Butterbrot etwas ganz normales. Es besteht aus einer Scheibe Brot, Butter und einem Belag, der möglichst schmackhaft sein sollte. So ein Butterbrot kommt uns gerade recht, um den Unterschied zwischen Betriebssystem und Programm zu verdeutlichen.

Was Sie immer brauchen, ist die Scheibe Brot. Auf den Computer übertragen, ist das der Computer selber. Weil das Butterbrot eben Butterbrot heißt, brauchen Sie auch unbedingt Butter drauf (außerdem quietscht es dann nicht so beim Essen). Übertragen auf den Computer: Wenn Sie einen Computer haben und irgendetwas damit anfangen wollen, brauchen Sie ein Betriebssystem. Was Sie auf Ihr Butterbrot legen, ist Ihre Sache und richtet sich nach Ihrem Geschmack oder Ihrem Hunger. Genauso beim Computer. Welches Programm Sie verwenden, um auf dem Bildschirm zu spielen oder Texte zu verarbeiten oder sonstwas zu machen, liegt an Ihrem persönlichen Bedarf und Geschmack.

Spätestens jetzt werden Sie erkennen, daß die Frage weiter oben falsch gestellt ist. Denn: Wenn der Computer vorhanden ist und ein Betriebssystem vorhanden ist, dann läuft jedes Programm, das für diese Art von Computer, unter diesem Betriebssystem geschrieben wurde.

II.1 Was ist ein Programm?

Grundsätzlich besteht jedes Programm aus nacheinander aufgeschriebenen Befehlen, die vom Computer in der Reihenfolge abgearbeitet werden. Einer nach dem anderen. Der Computer braucht die Befehle in einer Sprache, die er versteht und verarbeiten kann. Weil er selber eine Maschine ist, heißt diese Sprache "Maschinensprache". Programme, die in Maschinensprache geschrieben sind, bleiben für "normale" Menschen unverständlich. Aber der Computer freut sich. Weil diese Sprache dem Computer quasi auf den Leib geschrieben ist, arbeitet er die Befehlsfolgen auch besonders schnell ab.

Für "normale" Menschen gibt es zur Arbeitserleichterung noch die sogenannten Hochsprachen oder Programmiersprachen. Sie kennen sicherlich einige davon. Besonders bekannt ist BASIC oder auch COBOL oder FORTRAN oder PASCAL oder ganz neu auch MODULA 2. Bei diesen höheren Programmiersprachen werden für Menschen verständliche Wörter als Kommandos benutzt, innerhalb des Computers in Maschinensprache übersetzt und vom Computer abgearbeitet.

Ein Programm ist eben nichts anderes als eine Folge von Befehlen, die dem Computer genau sagen, Schrittchen für Schrittchen, was er wann zu tun hat. Die Länge eines solchen Programms kann zwischen zwei Befehlszeilen und fast unendlich vielen Befehlszeilen liegen. Das kommt einfach nur noch auf die Speicherkapazität im Hauptspeicher des Computers und auf den Laufwerken an.

Nun reicht es aber nicht, daß ein Programm die Anforderungen des Benutzers erfüllt, zum Beispiel Textverarbeitung ermöglicht. Es muß auch intern, innerhalb des Computers, eine ganze Reihe von Nebenaufgaben wahrnehmen, damit die Sache überhaupt klappt. Zum Beispiel muß es feststellen, welche Taste auf der Tastatur gerade gedrückt wurde, welches Zeichen auf dem Bildschirm oder dem Drucker darzustellen ist, ob und wo auf dem Massenspeicher die gerade benötigten Daten oder Programme stehen oder Daten von der Diskette in den Arbeitsspeicher des Computers laden.

Weiter muß sichergestellt sein, daß die von der Diskette gelesenen Daten an die richtige Stelle im Arbeitsspeicher gelangen und daß auf der Diskette genügend Platz für neue Aufzeichnungen vorhanden ist. Außerdem muß ein Inhaltsverzeichnis auf der Diskette geführt werden, u.s.w., u.s.w.

II.2 Unterprogramme

Wie Sie sehen, eine ganze Menge Arbeit so quasi nebenbei. Und diese "Nebenbei-Arbeit" muß jedesmal für jedes Programm neu programmiert werden. Für jeden Teil eines Programms muß ganz klar festgelegt sein, was im Computer zu passieren hat. Erschwerend kommt hinzu, daß diese Aufgaben von Computertyp zu Computertyp, so ähnlich sie sich auch sehen mögen, jedesmal anders zu lösen ist. Ein klein bißchen anders als andere ist eigentlich jeder Computer. Würde man nun ein längeres Programm quasi in einem Stück schreiben, müßte es für jeden Computertyp umgeschrieben werden, um vernünftig abzulaufen.

Diese unnötige Arbeit kann man sich sparen, wenn man das Programm in ein Hauptprogramm und mehrere Unterprogramme aufteilt. Die Unterprogramme werden nur dann in den Arbeitsspeicher des Computers geladen, wenn sie wirklich gebraucht werden.

Um das Programm dann an einen anderen Rechnertyp anzupassen, nimmt man einfach das eine Blöckchen Unterprogramm weg, schreibt ein neues dazu, und schon läuft das gesamte Programm auf dem neuen Computer ohne Einschränkung. Ein weiterer Vorteil dieser Unterprogramm-Technik ist, daß das Hauptprogramm gar nicht mehr weiß, wie eine bestimmte Teilaufgabe gelöst wird. Es erteilt einfach einen Befehl, zum Beispiel "Textausgeben", und das Unterprogramm erledigt diese Arbeit. Ob der Text auf dem Bildschirm, einem Drucker oder über einen Fernschreiber ausgegeben wird, kann dem Hauptprogramm völlig gleichgültig sein. Es erteilt einfach nur den Befehl und die Arbeit wird erledigt.

Diese Methode funktioniert, wenn im Hauptprogramm die Übergabe der Daten an die Unterprogramme standardisiert ist. Das kann man sich vorstellen wie einen Staffellauf, wo immer an

genau vorherbestimmten Stellen der ablösende Läufer steht und den Stab übernimmt. In der Computersprache heißt eine solche genormte und genau festgelegte Übergabestelle eine Schnittstelle. Dieser Ausdruck ist durchaus bildhaft zu verstehen, weil man ein Unterprogramm einfach abschneiden und ein neues dransetzen kann, ohne daß die Funktion des Hauptprogrammes darunter leidet oder es diese Veränderung überhaupt bemerkt. Es paßt alles wieder genau aufeinander und funktioniert reibungslos.

Nun ist es ja eigentlich Quatsch, daß jeder Programmierer bei jedem Programm immer wieder aufs Neue die internen Operationen in sein Programm mit aufnimmt. Das macht viel Arbeit, kostet eine Menge Zeit und Geld. Diese Idee hatten die Mannen von DIGITAL RESEARCH auch und machten sich daran, ein Standardprogramm für Mikro-Computer zu schreiben. Sie faßten die häufigsten Unterprogramme, die zur internen Steuerung nötig sind, zusammen, definierten die Schnittstellen und die Art der Datenübergabe zwischen Haupt- und Betriebsprogramm und hatten damit ein gemeinsames Betriebssystem für die verschiedensten Rechnertypen geschaffen.

Einsetzbar ist dieses Betriebssystem natürlich nur auf Computern mit ähnlichen oder gleichen Zentraleinheiten, weil ja schließlich die Maschinenbefehle des Programms verstanden und abgearbeitet werden müssen. Das Betriebssystem, das als erstes die vielen internen Arbeitsoperationen standardmäßig zur Verfügung stellte heißt CP/M. Diese Abkürzung steht für "Control Program for Micro-Processors", was in deutsch schlicht und einfach Steuerprogramm für Mikro-Prozessoren heißt. Dieses Betriebssystem arbeitet mit Mikro-Prozessoren vom Typ 8080, 8085 oder Z80.

II.3 Die Aufgaben von CP/M

Im wesentlichen erledigt CP/M die folgenden Grundaufgaben: Eingabe von Zeichen, Ausgabe von Zeichen, Verwalten des verfügbaren Speicherplatzes auf dem Massenspeicher, Lesen von Disketten-Aufzeichnungen und Schreiben neuer Aufzeichnungen auf Diskette oder in den Massenspeicher. Diese Dienstroutinen können von allen Anwenderprogrammen in einheitlicher und standardisierter Weise benutzt werden.

Damit das klappt, muß CP/M aber zwei verschiedene Arten von Arbeit durchführen, weshalb es in zwei große Blöcke aufgeteilt ist: Der erste Teil, BDOS (Basic Disc Operating System = Grundbetriebssystem für Disketten) kümmert sich um alle Aufgaben, die unabhängig von dem jeweiligen Computersystem immer in der selben Art und Weise zu lösen sind. Der zweite Teil des Betriebssystems, das sogenannte BIOS (Basic I/O-System = Grundsystem zur Ein- und Ausgabe) enthält all die Programmteile, die für ein bestimmtes Computer-System angepaßt und notwendig sind.

Jedes Computermodell hat nämlich so seine eigenen Ansichten und Methoden, wie bestimmte Aufgaben erledigt werden, und man kann nicht einfach ein Betriebssystem von einem Gerät nehmen und es auf einem anderen laufen lassen. Bevor solch ein übernommenes Betriebssystem funktioniert, muß das BIOS auf die neue Maschine angepaßt werden. Normalerweise sollten Sie bei Ihrem Computer damit keine Schwierigkeiten haben, denn jeder Computer wird mit einem angepaßten und passenden Betriebssystem ausgeliefert (hoffentlich).

II.4 CP/M und die verschiedenen Versionen

Jedes Programm, das sich einige Zeit auf dem Computermarkt behaupten kann, erfährt auch Verbesserungen und kommt danach in einer neuen Version auf den Markt. Es gibt nun mal keine fehlerfreien Programme, und einige dieser Fehler werden erst entdeckt, wenn ein Programm in vielfältigem Einsatz ist und wirklich alle Möglichkeiten ausprobiert werden, die es überhaupt bietet.
Treffen genügend Fehlermeldungen bei dem Hersteller des Programms ein, verbessert er die Fehler und entschließt sich irgendwann, eine neue Version des Programms auf den Markt zu bringen. Als nahezu fehlerfrei kann CP/M in der Version 2.2 gelten, die praktisch das Standardbetriebssystem für alle 8-Bit-Rechner ist. Die neuere Version ,CP/M 3.0, war also keine Neuausgabe um alte Fehler auszumerzen, sondern erfuhr eine deutliche Leistungssteigerung und erhielt zusätzliche, neue Befehle. Außerdem mußte CP/M an die immer leistungsfähiger werdenden Computer und Computertechniken angepaßt werden.

II.5 Der CP/M-Prompt

So, ich glaube das reicht. Lassen wir mal die Theorie da, wo sie hingehört und wenden uns dem wirklichen Leben, sprich dem Computer und seinem Betriebssystem zu. Was Sie als nächstes brauchen, ist ein funktionsfähiger Computer, ein Floppy-Laufwerk und eine Diskette mit dem CP/M-Betriebssystem drauf. Ist alles vorhanden, schalten Sie Ihren Computer ein, schieben die Diskette mit dem CP/M-Betriebssystem in Laufwerk "A" oder Laufwerk "1" und verriegeln es. Diese Vorgehensweise sollten Sie sich merken. Schalten Sie nämlich den Computer ein oder aus, während eine Diskette im Laufwerk

liegt, kann es durch den Stromstoß zu ungewollten Bewegungen des Schreib/Lesekopfs kommen, und Ihre wertvolle Diskette ist nicht mehr zu gebrauchen.

Also - Sie haben jetzt Ihren Computer eingeschaltet, Ihre CP/M-Systemdiskette eingelegt und den Startvorgang entsprechend der Beschreibung in Ihrem Handbuch vorgenommen. Beim Schneider CPC geht das so, daß Sie das Gerät (aus- und wieder) einschalten und dann das Kommando:

ICPM

eingeben. Den netten senkrechten Strich erhalten Sie, indem Sie die Tasten SHIFT + @ drücken.

Das Diskettenlaufwerk beginnt zu rattern und klickern, die Funktionslampe leuchtet auf, und kurz darauf sehen Sie auf dem Bildschirm eine Meldung. Diese sogenannte Anfangsmeldung sieht bei jedem Computer etwas anders aus. Das liegt daran, daß diese Meldung vom BIOS-Teil des Betriebssystems, also jenem Teil, der auf jeden Computer extra angepaßt werden muß, ausgegeben wird. Jeder Computerhersteller paßt sich diesen Teil für seine Maschine an und gestaltet die Anfangsmeldung nach seinen eigenen Vorstellungen. Eine übliche Anfangsmeldung bei CP/M sieht so aus:

CPM 2.2 - AMSTRAD Consumer Electronics plc.
A>

Beim CPC 6128 sieht diese Meldung noch ein wenig anders aus, da es sich hier um die Einschaltmeldung von CP/M 3.0 handelt.

CP/M Plus - AMSTRAD Consumer Electronics plc.
V 1.0, 61K TPA, 1 (2) Disc drive

Die Anfangsmeldung bedeutet für Sie, daß das CP/M-Betriebssystem richtig in den Arbeitsspeicher des Computers geladen wurde. Direkt anschließend meldet sich CP/M mit seiner Bereitschaftsmeldung, dem sogenannten Betriebssystem-Prompt. Dieser Prompt sieht so aus:

A>

Das große "A" sagt dem Benutzer, daß er mit dem Laufwerk "A" oder dem Laufwerk "1" seines Systems arbeitet, das ">" - Zeichen ist die eigentliche Bereitschaftsmeldung von CP/M. In der Zeile hinter der Bereitschaftsmeldung wartet CP/M auf die Eingabe eines Befehls. Na, dann wollen wir CP/M mal nicht so lange warten lassen und schreiben:

A>ABCDEFGH

Nun steht der Cursor hinter dem letzten Buchstaben, und nichts rührt sich, still ruht der See. Grund: CP/M weiß nicht, ob wir mit der Befehlseingabe bereits fertig sind oder vielleicht noch was dazuschreiben wollen. Um die Eingabe abzuschließen, müssen wir CP/M noch mitteilen, daß wir mit der Eingabe fertig sind.

Die Eingabe einer Zeile wird beim Computer wie bei der Schreibmaschine mit einem Wagenrücklauf beendet. Diese Taste heißt bei Computern häufig RETURN oder ENTER, was eine Abkürzung von CARRIAGE RETURN oder für "Eingabe" ist. Sobald Sie diese Taste drücken, weiß der Computer, daß Sie mit

Ihrer Befehlseingabe fertig sind und beginnt den eingegebenen Befehl auszuführen. Sie drücken jetzt auf die Wagenrücklauf- oder RETURN-Taste und sehen

A>ABCDEFGH
ABCDEFGH?
A>

Tja, und was ist passiert? CP/M hat die Zeile gelesen, den Befehl nicht verstanden, und sagt uns das. Das Betriebssystem meldet uns solche unverständlichen Befehle, indem es die Eingabe wiederholt und mit einem Fragezeichen versieht. Das Fragezeichen bedeutet dabei etwa soviel wie: "Was soll der Quatsch? Ich kann damit nichts anfangen."

Das war also nichts. Versuchen wir doch mal, ob es vielleicht an der Großschreibweise gelegen hat. Sie geben den gleichen "Befehl" wie eben, nur in Kleinbuchstaben ein:
A>abcdefgh

Als Antwort darauf erhalten Sie:

A>abcdefgh
ABCDEFGH?
A>

Das Betriebssystem hat den Befehl also auch in Kleinbuchstaben nicht begriffen, macht aber etwas anderes. Ist Ihnen aufgefallen, daß die Befehlszeile in Großbuchstaben wiederholt wurde, obwohl Sie sie in Kleinbuchstaben eingegeben haben? Das ist eine Eigenschaft von CP/M, die wir in manchen

Fällen sogar ganz bewußt beachten müssen. CP/M wandelt alle Buchstaben die Sie eingeben, erst einmal in Großbuchstaben um und bearbeitet sie dann.

Damit Sie sehen, daß CP/M wirklich auch arbeiten kann, wollen wir jetzt einmal einen Befehl eingeben, der auch etwas bewirkt. Weiter vorne haben Sie gelesen, daß es zu den Aufgaben des Betriebssystems gehört, die Einträge auf der Diskette in einem Inhaltsverzeichnis zu verwalten. CP/M legt zu diesem Zweck auf jeder Diskette ein Inhaltsverzeichnis (in englisch Directory) an und zeigt das auf Verlangen vor.

Wenn wir dieses Inhaltsverzeichnis betrachten wollen, müssen wir dem Betriebssystem irgendwie mitteilen, daß wir es sehen wollen und daß es dieses Inhaltsverzeichnis auf dem Bildschirm darstellen soll. Der Befehl besteht aus den drei Buchstaben DIR, was eine Abkürzung für Directory ist. Geben Sie diesen Befehl jetzt ein und anschließend ENTER:

A>DIR

Sie sehen das Inhaltsverzeichnis Ihrer Betriebssystem-Diskette auf dem Bildschirm. Schauen Sie es sich jetzt einmal genauer an. Ganz links wird das Laufwerk angegeben, in diesem Fall "A:". Daneben stehen die Namen von Dateien, und durch einen Leerraum getrennt, wird noch die Art der Datei angegeben. Sehr häufig sehen Sie den Zusatz COM, zum Beispiel bei DDT.COM, PIP.COM, ASM.COM. Diese Dateibezeichnung gibt Aufschluß über den Inhalt der Datei. Für uns sind im Moment die COM-Dateien besonders interessant, weil sie sofort ausführbare Programme enthalten.

COM ist die Abkürzung für das englische Wort "Command" oder Befehl. Die so bezeichneten Dateien enthalten Befehle, die

sofort ausgeführt werden können. Wenn Sie den Namen eines solchen Programms ohne den Anhang COM hinter den Betriebssystem-Prompt eintippen, wird das Programm direkt in den Arbeitsspeicher des Computers geladen und ausgeführt. Es genügt also, einfach PIP einzugeben, ein RETURN dazu, und schon startet das Programm.

II.6 Sicher ist sicher

Falls Sie jetzt noch immer mit Ihrer Original-Diskette arbeiten, müssen wir uns schleunigst darum kümmern, eine Sicherheitskopie zu erstellen. Sie sollten sich angewöhnen, nie mit den Originalen eines Programms zu arbeiten, sondern immer eine Kopie des Programms zu erstellen und das Original gut und sicher aufzubewahren. Wie sich das für ein Betriebssystem gehört, das dem Benutzer ja die Arbeit erleichtern soll, ermöglicht CP/M die Herstellung von Sicherheitskopien.

II.7 Was Sie jetzt schon wissen

* *Sie kennen eine einfache Erklärung, was ein Betriebssystem für Aufgaben zu bewältigen hat*

* *Ihnen ist klar, was ein Programm ist und wofür Unterprogramme da sind*

* *Sie kennen auch die Aufgaben von CP/M*

* *Sie wissen, was der CP/M-Prompt ist und wie Sie das Inhaltsverzeichnis Ihrer Disketten anzeigen können*

* *Sie haben sich fest vorgenommen, von jeder wichtigen Diskette mindestens eine Sicherheitskopie anzulegen*

III. Arbeiten mit CP/M

III.1 Die System-Diskette

Ich habe Sie bereits darauf hingewiesen, daß Sie von Ihren wertvollen Originaldisketten Kopien anfertigen sollten, damit im Falle eines Falles Ihr Original nicht beschädigt wird. Wie eine komplette Diskette kopiert wird, will ich Ihnen jetzt zeigen, und wir benutzen dazu CP/M-Befehle, die erst später genauer erklärt werden. Sie brauchen zwei Programme von der CP/M-Diskette, um eine komplette Diskette kopieren zu können. Wenn Sie noch einmal DIR eingeben und sich das Inhaltsverzeichnis genau ansehen, werden Sie unter anderem zwei Programme finden, die heißen:

SYSGEN.COM und PIP.COM bei CP/M 2.2 und

COPYSYS.COM und PIP.COM bei der Version 3.0

Allerdings sieht dies bei der Firma Schneider bezüglich CP/M 3.0 ein wenig anders aus. Die COM-Datei COPYSYS befindet sich auf keiner der drei Diskettenseiten. Das Programm zum Kopieren des Systems COPYSYS ist in dem Programmpaket DISCKIT3 integriert. Dieses Programmpaket ist zwar sehr komfortabel, aber mit Verlaub gesagt: Es wäre sicherlich nicht sehr aufwendig gewesen, die Datei COPYSYS.COM trotzdem mitzuliefern. Es fehlt auch die COM-Datei zum Formatieren einer Diskette. Ich beschreibe im folgenden Text nun den "normalen" Vorgang unter CP/M 3.0, d.h. wenn sich die beiden COM-Dateien mit auf den Systemdisketten befänden. Schlagen Sie in Kapitel 7 unter FORMAT und COPYSYS nach, wie Sie diese Kommandos auf dem Schneider anwenden können.

Mit diesen beiden Programmen machen Sie unterschiedliche

Sachen. Mit dem Programm SYSGEN (COPYSYS) kopieren Sie das CP/M-System auf die ersten beiden Spuren jeder Diskette. Deshalb heißen diese beiden Spuren auch die System-Spuren. Diese werden beim Starten Ihres Computers als erste gelesen, und das darauf stehende Programm in den Arbeitsspeicher Ihres Computers geladen. Auf die System-Spuren haben Sie als Computer-Benutzer keinen direkten Zugriff, weshalb Sie auch kein CP/M.COM oder etwas ähnliches auf der Diskette oder in Ihrem Inhaltsverzeichnis finden können. Sie brauchen nur zu wissen, daß die System-Spuren immer für CP/M reserviert sind und hier alle Programmteile stehen, die zum Betrieb eines Computers notwendig sind.

Wollen Sie nun aber eine Diskette komplett kopieren, d.h. alle Programme und auch das CP/M-Betriebssystem, brauchen Sie eine Möglichkeit, um auf diese beiden Systemspuren zugreifen zu können. Dazu verhilft Ihnen das Programm SYSGEN (CP/M 2.2) oder in CP/M 3.0 das Programm COPYSYS. Beides sind spezielle Programme, die es ermöglichen, den Inhalt der Systemspuren von einer Diskette auf die andere zu überspielen.

Bevor Sie sich nun an die Kopierarbeit machen, sollten Sie dafür sorgen, daß Sie eine frisch formatierte und leere Diskette zur Verfügung haben (Eine Diskette wird mit dem CP/M-Kommando FORMAT formatiert, indem Sie diesen Befehl hinter den CP/M-Prompt eingeben und den auf dem Bildschirm erscheinenden Anweisungen folgen).

III.2 Kopieren mit einem Laufwerk

Haben Sie alles parat, legen Sie Ihre CP/M-Originaldiskette in das Laufwerk A ein und schließen die Laufwerkstür. Arbeiten Sie mit nur einem Laufwerk, bekommen Sie ein wenige Mehrarbeit, als wenn Sie mit zwei Laufwerken operieren könnten. Aber es geht auch so, wenn Sie die folgenden Schritte befolgen. Um Ihr Betriebssystem zu kopieren, schieben Sie die Originaldiskette in das Laufwerk und geben ein:

A>SYSGEN (ENTER)

Daraufhin erscheint diese Meldung auf dem Bildschirm:

Please insert SOURCE disc into drive A
then press any key:

wobei mit SOURCE Ihre Originaldiskette gemeint ist oder die, von der das Betriebssystem genommen werden soll. Nachdem Sie die Diskette im Laufwerk haben, drücken Sie ENTER und diese Meldung erscheint:

Please insert DESTINATION disc into drive A
then press any key:

Sie geben daraufhin die neue Diskette in das Laufwerk, das nach einem erneuten ENTER seine Arbeit aufnimmt. Lange dauert das nicht und Sie sehen bald das:

Do you wish to reconfigure another disc (Y/N)?

Wenn Sie hier mit Y für Ja antworten, können Sie noch weitere Disketten mit dem Betriebssystem versehen. Bei einem klaren N für Nein, landen Sie wieder beim System-Prompt.

III.3 Kopieren mit zwei Laufwerken und CP/M 3.0

Wenn Sie mit zwei Laufwerken arbeiten, funktioniert der Vorgang genauso, nur daß Sie eben von einem Laufwerk zum anderen kopieren. Die leere Diskette legen Sie in Laufwerk B und schließen auch dort die Tür. Sie starten das Betriebssystem, tippen hinter die Bereitschaftsmeldung von CP/M COPYSYS (SYSGEN bei CP/M 2.2) ein und geben anschließend ein ENTER.

Die beiden Programme melden sich anschließend mit ihrem Namen und ihrer Versionsnummer und verlangen die Eingabe dès Laufwerks, von dem kopiert werden soll. Da Ihre CP/M-Betriebssystem-Diskette im Laufwerk A liegt, geben Sie ein "A" ein und werden gleich darauf gefragt, ob auch die richtige Diskette im Laufwerk A liegt. Wenn ja, drücken Sie auf ENTER und die Fragerei geht weiter.

Als nächstes sollen Sie das Laufwerk angeben, in dem Ihre jungfräuliche, sprich neu formatierte, Diskette liegt. Auch hier brauchen Sie wieder kein ENTER einzugeben, da das Programm sofort als nächstes wissen möchte, ob auch die Diskette eingelegt ist und zur Bestätigung ausdrücklich nach einem ENTER verlangt. Wenn Sie alle Instruktionen befolgt und das letzte ENTER eingegeben haben, beginnt das Laufwerk

B zu laufen und das CP/M-Betriebssystem wird auf die neue Diskette geschrieben. Danach kommt die Frage, ob CPM3.SYS auch kopiert werden soll. Das sind wichtige Hilfsprogramme und Sie sollten diese auf jeder Kopie Ihrer Systemdiskette haben. Sobald Laufwerk B sich wieder beruhigt hat, ist die Arbeit beendet und Sie könnten die nächste Diskette zum Übertragen des Betriebssystems einlegen. Da wir mit unserer Betriebsdiskette aber noch mehr vorhaben, drücken Sie einfach auf ENTER und beenden das Systemkopierprogramm.

III.4 Inhaltsverzeichnis ansehen

Sie haben etwas von A nach B kopiert und sind neugierig, was auf der B-Diskette alles steht. Damit Sie nicht vor Neugier platzen, geben Sie direkt hinter den CP/M-Prompt ein:

DIR B:

Wenn Sie die Leerstelle nach DIR nicht vergessen haben, werden Sie sehen, wie das Laufwerk B zu arbeiten beginnt, und Sie eine Meldung auf dem Bildschirm bekommen. Mit DIR B: haben Sie CP/M angewiesen, den Inhalt der Diskette im Laufwerk B anzuzeigen. Weil Sie aber bisher nur das Betriebssystem und noch keine Datei kopiert haben, erscheint jetzt die Meldung NO FILE. FILE wird im Deutschen mit Datei übersetzt und NO FILE bedeutet demnach nichts anderes, als daß nichts auf der Diskette steht. Wie Sie sich erinnern, werden die Programme der ersten beiden Systemspuren jeder Diskette versteckt. Sie haben darauf keinen direkten Zugriff und können sie auch nicht im Inhaltsverzeichnis sehen.

III.5 Kopieren mit PIP und zwei Laufwerken

Weil wir aber die gesamte CP/M-Systemdiskette und nicht nur das Betriebssystem kopieren wollen, bedienen wir uns jetzt eines anderen Programms. Dieses zweite Programm hat den unendlich langen, deutschen Namen: Prozessor für Datenaustausch zwischen Peripherie-Einheiten, was im Englischen zu Peripheral Interchange Processor wird, und abgekürzt ein lustiges PIP ergibt. Was nun dieses Programm macht, ist überhaupt nicht zum Piepen, sondern unerhört nützlich.

Sie haben immer noch die beiden Disketten in den Laufwerken, und tippen nun PIP und anschließend das ENTER ein. Der Computer lädt PIP nun in den Arbeitsspeicher, worauf sich dieses mit einem eigenen PROMPT meldet. Das sieht so aus:

```
A>PIP
*
```

Der Stern zeigt Ihnen an, daß PIP zur Aufnahme von Befehlen bereit ist. Dieses Programm ist ausgesprochen nützlich und leistungsfähig. Sie werden die vielen Möglichkeiten von PIP im Kapitel 6 genauer kennenlernen.

Bevor wir PIP nun einen entsprechenden Befehl geben, überlegen wir uns, was es eigentlich machen soll. Die Dateien von der Diskette in Laufwerk A sollen auf die Diskette in Laufwerk B übertragen werden. Man könnte auch sagen, "B:" bekommt alle Dateien, die auf "A:" gespeichert sind.

Den letzten Satz sollten Sie sich genau ansehen und vor allem gut einprägen. Er ist entscheident bei der Arbeit mit PIP. Wenn alle Dateien von A nach B übertragen sind, ist der

Inhalt beider Disketten gleich. Deshalb wird in PIP-Befehlen das Gleichheitszeichen benutzt. Bleibt nur noch die Schwierigkeit, wie wir PIP sagen, daß es alle Dateien nehmen soll und nicht nur irgendeine einzelne.

Für diesen Fall ist vorgesorgt. Das Sternchen anstelle eines Dateinamens bedeutet in PIP soviel wie "alle Dateien". Sie können das Sternchen wie einen Joker beim Kartenspiel verstehen, der für jede beliebige Karte eingesetzt werden kann. Weil sich Dateien aber einmal durch den Dateinamen, zweitens aber durch die Dateikennung, das sind die drei kleinen Buchstaben hinter dem Punkt, unterscheiden, müssen Sie fürs Kopieren aller Dateien ein Sternchen/Punkt/Sternchen eingeben. Das sieht auf dem Bildschirm so aus

B:=A:*.*

Weil wir sichergehen wollen, daß alle Dateien auch wirklich korrekt übertragen werden, lassen wir gleich jede Datei überprüfen, ob sie auch richtig rübergekommen ist. Wir können PIP diese Arbeit übertragen, indem wir ein "V" in eckigen Klammern hinter unseren Befehl schreiben. "V" steht für "Verify", was soviel wie nachprüfen, verifizieren heißt. Damit PIP diesen Befehl richtig versteht, muß er allerdings in eckigen Klammern stehen.
Haben Sie den kompletten Befehl eingegeben,

A>PIP
***B:=A:*.COM**

sehen Sie folgendes CP/M 3.0-Inhaltsverzeichnis auf dem Bildschirm:

```
A: CCP      COM : DATE     COM : DEVICE   COM : DIR      COM : DUMP     COM
A: ED COM : ERASE COM : GENCOM COM : GENCPM COM : GET COM
A: LIB COM : LINK COM : MAC COM : PATCH COM : INITDIR COM
A: PUT COM : RENAME COM : RMAC COM : SAVE COM : SET COM
A: HEXCOM COM : SETDEF COM : SHOW COM : SID COM : SUBMIT COM
A: TYPE COM : XREF COM : COPYSYS COM : FORMAT COM : HELP COM
A: PIP COM
```

Wie Sie während des Kopiervorgangs sehen, teilt Ihnen PIP mit, welche Datei gerade kopiert wird. Ist die Arbeit ge-
tan, meldet sich PIP wieder mit seinem Bereitschaftszeichen, dem Stern. Wenn Sie für PIP keine weitere Arbeit haben, und ich glaube, wir haben im Moment nichts mehr zu tun für PIP, geben Sie ENTER ein, und das Betriebssystem meldet sich wieder mit seinem A-Prompt.

Sollten Sie der Technik nicht so ganz trauen, können Sie mit DIR B: nachsehen, ob wirklich alle Dateien auf dem B:-Laufwerk angekommen sind. Ist alles zufriedenstellend, nehmen Sie die Diskette aus dem A-Laufwerk, also Ihre Original-Diskette, verfrachten Sie in die Disketten-Hülle und legen diese Diskette weit und gut gesichert weg. Sie arbeiten in Zukunft nur noch mit der Kopie, die Sie eben erstellt haben und ersparen sich viel Ärger, wenn mal was schiefgeht. Denn: Eine Diskette, um ein neues System zu kopieren, kostet Sie etwa fünf Mark (die 3-Zoll-Disketten für den Schneider sind etwas teurer), ein neues CP/M-Betriebssystem dagegen kostet gleich einige hundert Märker.

III.6 Regeln für Dateinamen

Bis jetzt haben Sie schon einige Male etwas über Dateien gehört, haben auch Datei-Namen im Inhaltsverzeichnis auf Ihrem Bildschirm gesehen, aber was eine Datei ist und was es damit auf sich hat, haben Sie noch nicht erfahren.

Wie Sie wissen, besitzt jede Datei in CP/M einen Namen. Das ist einmal wichtig für Sie, damit Sie wissen, was drinsteht, zweitens für CP/M wichtig, damit es diese Datei auf Ihren Aufruf hin auch finden kann. Leider haben die CP/M-Konstrukteure eine Beschränkung eingebaut, die zum Beispiel mir überhaupt nicht paßt.

Ein Datei-Name in CP/M darf aus maximal acht Zeichen bestehen, einem Punkt und wiederum drei Zeichen für die sogenannte Kennung. Das bedeutet, Sie müssen für Ihre Dateien Namen finden, die maximal acht Zeichen lang sind und trotzdem einen Sinn und einen Rückschluß auf deren Inhalt ergeben.
Die zulässigen Zeichen für einen Datei-Namen sind alle 26 Buchstaben des Alphabeths sowie das Plus-, Minus-, Schrägstrich-, Prozent- und Dollarzeichen in jeder Position des ersten Namens oder innerhalb der Kennung. Die Zeichen "kleiner als", "größer als", Punkt, Komma, Doppelpunkt, Gleichheitszeichen, Strichpunkt, Stern, Fragezeichen, eckige Klammer auf, eckige Klammer zu, Ausrufezeichen sollten Sie in Datei-Namen oder der Kennung nicht benutzen, da diese Zeichen eine bestimmte Bedeutung für CP/M haben und zu Fehlfunktionen führen können.

Aus meiner langjährigen Erfahrung mit CP/M und anderen Programmen kann ich Ihnen nur raten, Ihren Dateien immer einen sinnvollen Namen zu geben, der Rückschlüsse auf den Inhalt der Datei zuläßt. Besonders wenn Sie viele Dateien auf Ihrer

Diskette stehen haben, kommen Sie mit einem Namen wie XK2512AC.ABC überhaupt nicht mehr zurecht. Bei der Wahl Ihres Datei-Namens brauchen Sie ansonsten keine besondere Vorsicht walten zu lassen, weil zwei Dateien mit dem gleichen Namen und der gleichen Kennung nicht gleichzeitig auf einer Diskette stehen können. Allerdings können Dateien mit gleichem Namen und unterschiedlicher Kennung gleichzeitig auf einer Diskette stehen.

III.7 Datei-Kennung

Die Kennung eines Datei-Namens können Sie frei gestalten, d.h. Sie können jede Datei benennen, wie Sie wollen, sollten dabei aber auf CP/M-typische Abkürzungen achten. Eine Tabelle der in CP/M üblichen Abkürzungen und deren Bedeutung finden Sie am Ende dieses Kapitels. Dort finden Sie zum Beispiel Programme mit der Kennung COM. Das ist die Abkürzung für COMMAND und bedeutet, daß diese Programme sofort ausführbaren Befehlscode enthalten.
Sie erinnern sich, daß PIP und SYSGEN oder COPYSYS solche COM-Programme sind. Außerdem sollten Sie sich davor hüten, Dateien mit der Kennung BAK oder $$$ zu benennen. BAK steht für "back up", was soviel wie Sicherheitskopie heißt und wird vom CP/M-Editor benutzt, um Sicherheitskopien von Dateien zu bezeichnen. Die Datei mit den drei Dollarzeichen am Ende versteht sich als Zwischendatei, die von einem Programm angelegt und am Ende des Programmlaufs rigoros gelöscht wird. PIP zum Beispiel legt sich solche Zwischendateien an und löscht diese am Ende total von der Diskette. Also lieber Vorsicht.

Ungeschickt ist es auch, gleiche Datei-Namen zu verwenden, und nur die Kennung zum Zählen der Dateien zu verwenden. Beispiel: Sie haben einen Text und benennen die erste Datei

TEXT.1, die zweite Datei TEXT.2 und die dritte Datei TEXT.3. Was im Moment recht logisch aussieht, wird Sie nachher ärgern. Sobald Sie nämlich die erste Datei bearbeitet haben, wird aus Ihrer Original-Datei die neue Datei TEXT.BAK und die überarbeitete Version heißt jetzt TEXT.1. Nun überarbeiten Sie die Datei TEXT.2, und auch hier wird die Original-Datei zu TEXT.BAK und die überarbeitete Version zu TEXT.2.

Da bei CP/M keine zwei Dateien mit genau dem gleichen Namen und der gleichen Kennung auf einer Diskette stehen dürfen, hat das Betriebssystem kurzerhand Ihre erste BAK-Datei gelöscht und durch die zweite überschrieben. Somit haben Sie keine Möglichkeit mehr, den Text Ihrer ersten Original-TEXT.1-Datei nochmal anzusehen.

Um dieses Problem zu umgehen, benennen Sie Ihre erste Datei mit TEXT1.TXT, Ihre zweite Datei mit TEXT2.TXT und Ihre dritte Datei mit TEXT3.TXT.

Wenn Sie nun die Dateien überarbeiten, bekommen Sie für jede eine eigene Back-up-Datei und haben jederzeit die Möglichkeit, Ihren Originaltext wieder hervorzuholen.

III.8 Eine Datei wiederfinden

Wenn Sie Ihren Computer intensiv nutzen und viele verschiedene Dateien erstellen, stehen Sie möglicherweise bald vor einem neuen Problem. Sie finden die Dateien, die Sie suchen, nicht mehr auf Anhieb in Ihrem Inhaltsverzeichnis. Ihnen kann es beispielsweise aber recht nützlich sein herauszufinden, wieviel Dateien mit dem Namen TEXT.TXT auf der Diskette bereits stehen.

Sie haben zwei Möglichkeiten, nach diesen Dateien suchen zu lassen. Als Helfer dienen hier in beiden betriebssystem-Versionen der Stern und das Fragezeichen. Den Stern haben Sie schon als sogenannten Joker bei unserer Kopier-Operation mit PIP kennengelernt, das Fragezeichen arbeitet ähnlich. Während der Stern stellvertretend für alle Zeichen des Datei-Namens oder der Kennung steht, arbeitet das Fragezeichen als Ersatz für ein Zeichen an einer bestimmten Position.

Die Eingabe "Text?" zum Beispiel sucht und findet alle Dateien mit dem Namen "Text" oder mit dem Namen "Text" plus einem weiteren Zeichen. So würden auch alle drei Dateien "TEXT1", "TEXT2" und "TEXT3" gefunden. Sollte zusätzlich noch eine Datei "TEXT" auf unserer Diskette stehen, würde diese auch mitgefunden und angezeigt. Das liegt daran, daß CP/M für einen Datei-Namen immer acht Zeichen reserviert. Ist Ihr eingegebener Datei-Name kürzer, füllt CP/M die restlichen Plätze mit Leerzeichen auf. Dabei sind Leerzeichen genauso gültige Zeichen wie Buchstaben oder Ziffern.

III.9 Suchen mit Fragezeichen (?)

Da das Fragezeichen für jedes Zeichen gesetzt werden kann, werden nicht nur die Dateien "TEXT1", "TEXT2" und "TEXT3" gefunden, sondern auch die Datei "TEXT". Der Einsatz des Fragezeichens bleibt aber nicht auf ein Zeichen beschränkt. Sie können genauso gut bis zu acht Fragezeichen für den Datei-Namen eingeben und bis zu drei Fragezeichen für die Kennung. Das aber ist eine lästige Tipperei, die Sie sich mit der Eingabe des Sternchens für den Datei-Namen und eines Sternchens für die Kennung ersparen können. Dem Betriebssystem ist es völlig gleichgültig, ob Sie acht Fragezeichen

für den Datei-Namen oder ein Sternchen eingeben. Intern wird sowieso jedes Sternchen in acht Fragezeichen umgewandelt, bevor die Suche beginnt.

Sie können auch mit dem Sternchen genauer suchen, wenn Sie vor den Stern den oder die Anfangsbuchstaben der gesuchten Dateien schreiben. Geben Sie allerdings ein Sternchen und dann noch ein paar Buchstaben ein, funktioniert die Suche nicht. Diese Methode, das Eingeben von Datei-Namen mit Fragezeichen oder Sternchen, können Sie zusammen mit den Befehlen DIR, TYPE, STAT, FILECOPY und PIP verwenden.

III.10 Was Sie jetzt schon wissen

* *Sie können nun eine Systemdiskette kopieren und Ihr Original gut aufheben*

* *Mit dem Stern und den Fragezeichen als Ersatz für fehlende Buchstaben können Sie auch umgehen und sich das Leben leichter machen*
* *Sie wissen, daß man Dateien immer sinnvolle Namen geben soll, damit man sie auch wiederfindet*

IV. Eingebaute Befehle

Im letzten Kapitel haben wir ein paar Grundfunktionen des Betriebssystems CP/M kennengelernt. Das ist aber längst noch nicht alles. In diesem Kapitel werden wir uns genauer mit den eingebauten CP/M-Befehlen befassen, die da lauten: USER, DIR, DIRSYS(CP/M3.0), ERA, ERASE(CP/M3.0), REN, RENAME (CP/M 3.0) und TYPE.

Außerdem kümmern wir uns später noch um die sogenannten Transienten-Programme (das sind CP/M-Programme, die Sie im Inhaltsverzeichnis sehen können und die als COM-Dateien gespeichert sind). Zusätzlich werden Sie Ihre Kenntnisse über die Befehle DIR und PIP, die Sie in dem vorherigen Kapitel schon kennengelernt haben, erweitern.

IV.1 Befehle, Parameter und Optionen

Bevor ich Ihnen weitere CP/M-Befehle erkläre, sollen Sie noch den grundsätzlichen Unterschied zwischen einem Befehl, einem Parameter und einer Option kennenlernen. Ein Befehl ist ein Befehlswort, das CP/M anweist, eine bestimmte Aktion auszuführen. Ein Parameter ist in der Regel ein Datei-Name, der angibt, auf welche Datei sich der Befehl bezieht. Eine Option steht bei CP/M in eckigen Klammern und kann, wie der Name schon sagt, auch weggelassen werden. Wird aber eine Option mit eingegeben, verändert sie auf eine bestimmte Weise die Funktion des vorne eingegebenen Kommandos.

Ein Beispiel für eine Option haben Sie bereits gesehen, als wir mit PIP Ihre Systemdiskette kopiert haben. Die Option lautete "V", und veranlaßte PIP, alle kopierten Dateien auch zu überprüfen.

Jenachdem, um welchen Befehl es sich handelt, werden Parameter mit eingegeben oder nicht. Sollte irgendwo ein Parameter notwendig sein, und Sie haben keinen eingegeben, verlangt CP/M ausdrücklich die Eingabe des Parameters. Sie brauchen also keine Angst zu haben, daß etwas nicht funktioniert.

Bei der Eingabe von Befehlen und Parametern müssen Sie darauf achten, daß mindestens ein Leerzeichen zwischen dem Befehl, und dem Parameter steht. Und das ist auch der einzige Ort, wo eine Leerstelle auftauchen darf. Weder im Befehl, noch im Parameter, noch in der Option dürfen an irgendwelchen Plätzen Leerstellen auftauchen.

Normalerweise werden Sie nur einen oder zwei Datei-Namen nach einem Kommando als Parameter eingeben. Sie sind allerdings nicht gezwungen, sich auf so wenige Angaben zu beschränken. Wenn Ihnen genug einfällt oder es Ihre Anwendung verlangt, können Sie auch die ganze Zeile vollschreiben, mit Control E (^E) eine neue Zeile anfangen und dort weitermachen. CP/M wird diese beiden Zeilen als eine Befehlszeile lesen.

IV.2 Die eingebauten Befehle

Sobald CP/M von der Diskette in den Arbeitsspeicher im Computer geladen ist, stehen Ihnen zwei Sorten von Kommandos zur Verfügung. Die eingebauten Kommandos sind im Arbeitsspeicher Ihres Computers und können sofort und sehr schnell von dort abgerufen werden, um irgendwelche Aufgaben zu erfüllen. Die anderen Kommandos stehen als Datei auf Ihrer Diskette und Sie können sie auflisten, wenn Sie sich das Inhaltsverzeichnis ansehen. Diese Programme lassen sich wie

ganz normale Dateien kopieren oder löschen oder, wenn Sie genügend Kenntnisse haben, auch verändern.
Diese Aufteilung in CP/M wurde gemacht, weil die beiden Systemspuren auf jeder Diskette, die extra für CP/M reserviert sind, längst nicht genug Platz für alle Hilfsprogramme bieten. Für Ihre Arbeit mit dem Computer ist es aber unerheblich, ob Sie ein eingebautes Kommando oder ein transientes Kommando aufrufen. Verlangen Sie zum Beispiel mit dem Kommando PIP ganz besondere Aktionen, lädt CP/M automatisch die Datei PIP.COM in den Arbeitsspeicher und führt Ihren Befehl aus.

Den einzigen Verlust, den Sie haben, ist etwas Zeitverlust durch das Lesen der Datei von der Diskette. Insgesamt besitzt das Betriebssystem CP/M 2.2 fünf und die Version 3.0 mittlerweile sechs eingebaute Kommandos, wobei hier vier eine Erweiterung durch eine gleichnamige COM-Datei erfahren. Wenn Sie in CP/M 3.0 eins dieser Kommandos aufrufen, müssen Sie nicht jedesmal den kompletten Namen eingeben. Sie können die Befehlseingabe so abkürzen, wie in der unten stehenden Tabelle:

Befehl	Abkürzung	Funktion
DIR	DIR	Zeigt das Inhaltsverzeichnis
DIRSYS	DIRS	Zeigt Verz. der SYSTEM-Dateien
ERASE	ERA	Löscht Dateien
RENAME	REN	Dateinamen ändern
TYPE	TYP	Zeigt Text-Datei
USER	USE	Ändert Benutzer-Bereich

In der 2er-Version kommen Sie mit den ausgeschriebenen Namen der Befehle nicht weiter, weil das Betriebssystem sie nicht versteht. Hier können Sie nur mit den Abkürzungen arbeiten.

Genaugenommen gibt es noch ein eingebautes Kommando in CP/M, das aber keinen eigenen Namen besitzt. Sie können dieses Kommando benutzen, um von Laufwerk A: auf Laufwerk B: oder auf irgendein anderes Laufwerk umzuschalten. Wenn Sie statt Laufwerk A lieber Laufwerk B als angemeldetes Laufwerk haben wollen, tippen Sie einfach ein:

B: (RETURN)

und CP/M macht Ihr B-Laufwerk zum angemeldeten Laufwerk.

Wenn Sie nun auf dem B-Laufwerk arbeiten, während Ihre CP/M-Systemdiskette im Laufwerk A liegt, können Sie trotzdem auf die CP/M-Programme auf der A-Diskette zugreifen. Dazu geben Sie einfach die Laufwerks-Bezeichnung plus Doppelpunkt vor dem Befehl ein. Das kann so aussehen:

B>A:DIR

Genauso können Sie den Laufwerks-Buchstaben auch vor einen Parameter setzen

B:DIR A:DATEI.XXX

IV.3 USER

Dieses eingebaute Programm ermöglicht es Ihnen, ein Speicher-Medium in 16 Bereiche zu unterteilen. Die Bezeichnung der Bereiche geht dabei von 0 bis 15. So eine Aufteilung kann recht nützlich sein, wenn sich mehrere Benutzer einen Computer teilen und ihre Dateien nicht durcheinander kommen sollen. Eine andere Möglichkeit ist, verschiedene Programme und die dazugehörigen Dateien in jeweils einem anderen Benutzer-Bereich aufzubewahren.

Zum Beispiel könnten Sie einen Bereich für Ihre Textdateien, einen Bereich für Ihre BASIC-Dateien, einen Bereich für geschäftliche Briefe u.s.w., u.s.w. anlegen. Richtig sinnvoll wird die Unterteilung eines Speicher-Mediums eigentlich aber erst bei Verwendung einer Hard-Disk, die erheblich mehr Platz als eine Diskette bietet und die Verwaltung aller Dateien schon schwierig macht.

Durch die Unterteilung in Bereiche ist es auch möglich, daß zwei Dateien mit genau dem gleichen Namen und der gleichen Kennung auf einem Speicher-Medium stehen. Allerdings - jede Datei in ihrem eigenen Bereich.

IV.3.1 USER-Bereiche bei CP/M 2.2

Aus dem Mehrplatzbetriebssystem MP/M wurde die Möglichkeit der verschiedenen Benutzerbereiche in die Version CP/M 2.2 übernommen. Jeder Bereich arbeitet wie eine selbstständige Diskette. Das heißt: Dateien werden in jedem Benutzerbereich einzeln erstellt und die benötigten Programme müssen alle in diesem Benutzerbereich zur Verfügung stehen. Zwischen den verschiedenen Benutzerbereichen wird mit dem Befehl:

USER n

umgeschaltet. Das "n" steht dabei für eine Ziffer zwischen 1 und 15. Sie können innerhalb eines Bereiches voll arbeiten, ohne die Dateien in anderen Bereichen zu berühren. Zum Beispiel löscht der Befehl:

ERA *.*

alle Dateien, aber nur innerhalb eines Benutzerbereiches. Arbeiten Sie mit zwei Laufwerken und kopieren Dateien von einem zum anderen Laufwerk, erscheinen diese Dateien normalerweise auf dem anderen Laufwerk im selben Benutzerbereich. Wenn Sie also von Bereich 3A mit PIP auf B: kopieren, landen die Dateien in dem Bereich 3B.

Nach jedem Kaltstart von CP/M beginnt das Betriebssystem im Benutzerbereich Null. Möchten Sie in einem anderen arbeiten, müssen Sie zuerst den neuen Benutzerbereich eingeben. Leider sehen Sie am CP/M-Prompt nicht, in welchem Benutzerbereich Sie sich gerade befinden. Das können Sie nur mit dem Befehl STAT herausbekommen.

IV.3.2 USER-Bereiche bei CP/M 3.0

Eine besondere Stellung nimmt hier der Benutzer-Bereich Null(0) ein. Programme und Dateien, die in diesem Bereich stehen, und zu System-Dateien erklärt wurden, können von jedem Benutzer-Bereich aus aufgerufen und verwendet werden. Genaueres dazu erfahren Sie in einem späteren Abschnitt. Sobald Sie sich in einem Benutzer-Bereich befinden und dort eine neue Datei erstellen, merkt sich CP/M automatisch, in welchem Benutzer-Bereich diese Datei erstellt wurde.

Jedesmal, wenn Sie Ihren Computer einschalten und CP/M geladen haben, sind Sie im Benutzer-Bereich 0. Wenn Sie in einem anderen Benutzer-Bereich arbeiten wollen, können Sie den Befehl USER eingeben. Angenommen, Sie wollen in Benutzer-Bereich Nummer 3, dann geben Sie ein:

USER 3

Achten Sie darauf, ein Leerzeichen hinter USER und der entsprechenden Zahl einzugeben. Sobald CP/M auf den neuen Benutzer-Bereich umgeschaltet hat, sehen Sie, daß sich etwas auf dem Schirm verändert. Der CP/M-Prompt ist nun nicht mehr einfach ein "A", sondern Sie sehen die Angabe des Benutzer-Bereiches vor dem A, in unserem Fall also eine "3". Nun wollen Sie sicherlich auch mit dem zweiten Laufwerk arbeiten und dort die Benutzer-Bereiche umschalten können. Um zum Beispiel in den Benutzer-Bereich 3 auf dem Laufwerk B zu gelangen, geben Sie ein

USER 3B:

und sehen dann den CP/M-Prompt, der so aussieht:

3B>

Wenn Sie jetzt den Befehl DIR eingeben, um sich das Inhaltsverzeichnis Ihrer Diskette anzusehen, sehen Sie "NO FILE". Vorausgesetzt natürlich, daß Sie in diesem Benutzer-Bereich noch keine Datei erstellt haben. Sie können von jedem Benutzer-Bereich aus auf alle eingebauten CP/M-Befehle zugreifen. Um wieder in den Benutzer-Bereich 0 zu kommen, geben Sie ein:

USER 0

und landen wieder ganz unten im untersten Benutzer-Bereich. Programme, die hier stehen und für alle Benutzer-Bereiche verfügbar sein sollen, können Sie mit dem SET-Befehl darauf vorbereiten. Wie das funktioniert, zeige ich Ihnen in einem späteren Abschnitt. Den Wechsel zwischen den Benutzer-Bereichen können Sie aber noch einfacher gestalten, indem Sie Ihren Änderungswunsch so eingeben:

B1:

Bei den Benutzer-Bereichen sollte noch eins beachtet werden: Wenn Sie mit PIP eine Datei von "A" nach "B" oder umgekehrt kopieren, wird sie nur im selben Benutzer-Bereich kopiert.

IV.4 DIR

Den Befehl DIR haben Sie schon kennengelernt, als Sie sich im vorherigen Kapitel das Inhaltsverzeichnis Ihrer Disketten ansahen. Der DIR-Befehl ist bei beiden CP/M-Versionen gleich, außer daß er in der 3er Ausführung aus zwei Programmen besteht. Einmal aus dem eingebauten Kommando DIR, was dem von CP/M 2.2 gleicht, zweitens aus dem Programm DIR.COM, das für erheblich erweiterte Hilfsfunktionen sorgt.

Wir beschäftigen uns jetzt erst einmal mit dem Befehl DIR, so, wie er in beiden Versionen benutzt werden kann.

Geben Sie das Kommando DIR ohne irgend etwas anderes dazu ein, erscheinen alle Dateien Ihrer Diskette in diesem speziellen Benutzer-Bereich. Wenn Sie also DIR im Benutzer-Bereich 3 befehlen, erscheinen auch nur die Dateien, die im Benutzer-Bereich 3 abgelegt sind. Manchmal stehen bei CP/M 3.0 in einem Benutzer-Bereich so viele Dateien, daß längst nicht alle auf einen Bildschirm passen.

In diesem Fall hält die Bildschirmausgabe des Inhaltsverzeichnisses an, bis Sie sich einen Überblick verschafft haben und durch Drücken irgendeiner Taste den Befehl zum Weitermachen geben. Möchten Sie die Ausgabe des Inhaltsverzeichnisses stoppen, drücken Sie einfach Control C (^C).

Arbeiten Sie gerade auf Laufwerk A, möchten aber sehen, ob eine bestimmte Datei im Laufwerk B vorhanden ist, gibt es zwei Möglichkeiten, das Inhaltsverzeichnis von B zu sehen. Sie tippen ein:

B:
DIR

und sehen das Inhaltsverzeichnis von Laufwerk B. Die zweite Methode geht schneller. Sie befehlen:

DIR B:

und sehen ebenfalls das Inhaltsverzeichnis von Laufwerk B auf Ihrem Bildschirm. Zusätzlicher Vorteil der zweiten Version: Sie bleiben auf Laufwerk A. Möchten Sie das Inhaltsverzeichnis Ihrer Diskette lieber ausdrucken, geben Sie ein:

DIR B:

dann Control P (^P) und ENTER. Nun wird Ihr komplettes Inhaltsverzeichnis auf dem Drucker ausgedruckt. Nach Beendigung des Druckes drücken Sie nochmal Control P, und der Drucker stellt die Arbeit ein.

IV.4.1 DIR mit Parametern

Die einfachen Funktionen von DIR sind eigentlich nicht besonders aufregend und man erwartet so etwas auch von einem normalen Betriebssystem. Richtig nützlich wird DIR aber, wenn Sie die vielen zusätzlichen Möglichkeiten nutzen, die dort drin stecken. Angenommen, Sie besitzen jede Menge Einträge in Ihrem Inhaltsverzeichnis und möchten eine bestimmte Datei heraussuchen. Statt nun alle Dateien einzeln durchzulesen und zu gucken, wo Ihre Datei steckt, geben Sie einfach ein

DIR B:TEXT.TXT

und bekommen diesen Namen angezeigt, wenn die Datei im Laufwerk B wirklich existiert. Ist die Datei nicht vorhanden, bekommen Sie von CP/M eine Fehlermeldung.

Wenn Sie die erweiterten Funktionen von DIR benutzen wollen, müssen Sie bei 3.0 mit dem Programm DIR.COM arbeiten. Das bedeutet: Wenn Sie auf Laufwerk B arbeiten, aber DIR.COM auf Laufwerk A zu finden ist, dürfen Sie nicht vergessen, die Laufwerksbezeichnung vor DIR mit einzugeben. Das sieht zum Beispiel so aus:

A:DIR [Optionen]

Sie erinnern sich sicher, daß das Sternchen uns eine Menge Tipparbeit beim Suchen abnehmen kann. Mit dem Befehl DIR und der Kombination mit dem Sternchen können Sie sich eine bestimmte Art von Dateien auflisten lassen. Das funktioniert so:

DIR *.COM

Wenn Sie den Befehl so eingeben, werden Sie auf Ihrem Bildschirm alle Dateien sehen, die als Kennung ein COM haben.

IV.4.2 Erweitertes DIR bei CP/M 3.0

Sie können in der neueren CP/M-Version genauso gut nach mehreren verschiedenen Datei-Typen suchen. Der entsprechende Befehl sieht so aus

DIR *.COM *.SUB

Damit das wirklich funktioniert, müssen Sie von CP/M ausdrücklich verlangen, daß DIR.COM benutzt wird.

Sie erreichen dies, indem Sie die Laufwerks-Bezeichnung wie A: oder B: vor dem DIR eingeben oder indem Sie eine Option in eckigen Klammern hinter die Datei-Namen schreiben. Die beiden, in diesem Fall möglichen, Optionen heißen FULL oder DIR.

IV.4.3 DIR und seine Optionen

Insgesamt können Sie mit DIR 18 verschiedene Optionen angeben. Damit ist DIR ähnlich leistungsfähig wie PIP und gehört zu den besonders nützlichen CP/M-Programmen. Normalerweise werden Sie wohl selten mehr als ein oder zwei Optionen gleichzeitig eingeben. Die Optionen zu DIR stehen immer in eckigen Klammern.

Wenn mehr als eine Option angegeben wird, werden die beiden durch Kommata oder einen Leerraum getrennt. Ist die Bezeichnung der Option eindeutig, können Sie den Namen der Option auch auf zwei Buchstaben abkürzen. Zur Vereinfachung können Sie die geschlossene eckige Klammer, auch weglassen, wenn es das letzte Zeichen in der Befehlszeile ist.

IV.4.4 DIRSYS

Wenn Sie sich in CP/M 3.0 ein Inhaltsverzeichnis ansehen, haben Sie sicherlich schon bemerkt, daß am unteren Rand des Bildschirms eine Meldung erscheint, die so aussieht

SYSTEM FILE (S) EXIST

Damit weist Sie CP/M darauf hin, daß außer den aufgelisteten Dateien noch sogenannte Systemdateien auf der Diskette stehen. Systemdateien sind solche Dateien, die im Benutzer-Bereich 0 liegen und von jedem Benutzer-Bereich aus zugänglich und verwendbar sind. Möchten Sie sich Ihre Systemdateien ansehen, geben Sie das Kommando DIRSYS oder abgekürzt einfach DIRS ein. Unter den aufgelisteten Dateien bekommen Sie dann die Meldung, daß "Nicht-Systemdateien" existieren.

Selbstverständlich haben Sie mit dem Kommando DIRSYS die gleichen Möglichkeiten wie mit dem Befehl DIR, Sie können also auch das Sternchen und die Fragezeichen benutzen, genau wie vorher besprochen.

IV.5 ERASE

Der Platz auf Ihrer Diskette und auch die Zahl der möglichen Einträge in Ihrem Inhaltsverzeichnis pro Diskette sind limitiert. Irgendwann werden Sie keinen Platz mehr auf der Diskette haben, und sind gezwungen eine Datei zu löschen oder auch mehrere. Das Betriebssystem CP/M bietet die Möglichkeit, Dateien mit dem Befehl ERA zu löschen.

Den Befehl geben Sie so ein:

ERASE D:NAME

wobei D für die Laufwerksbezeichnung steht und NAME für den Namen Ihrer Datei. Arbeiten Sie zum Beispiel mit Laufwerk A und wollen dort eine Datei löschen, brauchen Sie die Laufwerksbezeichnung nicht mit einzugeben.
Um mehrere Dateien auf einmal zu löschen, können Sie natürlich auch wieder die beiden Zeichen Sternchen und Fragezeichen benutzen. Allerdings sollten Sie genau überlegen, was Sie tun, wenn Sie diese gewaltigen Befehle benutzen. Nur zu leicht kann es Ihnen nämlich passieren, daß Sie die falschen Dateien löschen und sich nachher vor Wut in den Bauch beißen.

Möchten Sie zum Beispiel alle Dateien löschen, die ein BAK als Kennung besitzen, geben Sie ein:

ERA *.BAK

und sind bei CP/M 2.2 alle Dateien mit der entsprechenden

Kennung los. Bei Backup-Dateien, wie hier beschrieben ist ein Irrtum sicherlich nicht so schlimm. Stellen Sie sich aber bitte einmal vor, Sie hätten diesen Befehl gegeben:

ERA *.*

und dann, ohne nachzudenken ENTER hinterher. Tja, das war's dann auch schon. Ihre Daten sind futsch.

Weil dieser Befehl aber so mächtig ist und vieles auch zerstören kann, haben die CP/M-Erfinder eine klitzekleine Sicherheitsstufe eingebaut. Sobald Sie nämlich mit zwei Sternchen löschen (*.*), werden Sie sicherheitshalber nochmal gefragt, ob Sie das auch ernst meinen:

ALL (Y/N)

Soll alles gelöscht werden, geben Sie ein Y für Ja und dann ENTER ein. Andernfalls ein N für Nein und die Situation ist gerettet.

IV.5.1 Löschen mit ERA in CP/M 3.0

Wahrscheinlich hat der Chef von DIGITAL RESEARCH aus Versehen einmal selber alle seine Dateien mit seinen Geheimkonten gelöscht. Denn in der neuen CP/M-Version fragt das Programm auf die Eingabe mit einem Joker:

ERA *.BAS

schon nach, ob wirklich das alles gelöscht werden soll:

ERASE *.BAS (Y/N)?

Bevor Sie hier mit "Y" für Ja antworten, sollten Sie wirklich noch einmal ganz genau überlegen, was Sie tun. Sind Sie ganz sicher, daß der Befehl richtig formuliert ist und daß Sie genau diese Art von Dateien auch löschen wollen? Haben Sie erst mal Ihr "Y" eingetippt, gibt es keinen Weg zurück. Die Dateien sind weg. Achten Sie vor allem auch auf Schreibfehler. Zum Beispiel kann man leicht den Zusatz BAS durch einen Tippfehler als PAS schreiben. Wenn Sie diesen Befehl so losschicken, verschwinden alle PASCAL-Datein die auf Ihrer Diskette stehen...

Um sich vor solch unliebsamen Überraschungen zu schützen, gibt es zwei verschiedene Möglichkeiten. Einmal können Sie Ihre wichtigen Dateien "schreibschützen". Wie das geht, sehen Sie später. Zweitens können Sie an das Kommando ERASE die Option CONFIRM anhängen. CONFIRM heißt soviel wie bestätigen, und Sie können die Option abkürzen zu einem kurzen "C". Wenn Sie den Befehl nun so eingeben:

ERASE *.BAK[C

werden alle "back-up"-Dateien gelöscht, aber CP/M fragt vor jeder Löschung nach, ob diese Datei auch gelöscht werden soll.

IV.6 Dateinamen ändern mit REN(AME)

Sie haben bisher gesehen, daß wir uns auf Dateien beziehen, indem wir den Datei-Namen eingeben oder eintippen. Nun kann es ja sein, daß Ihnen ein Datei-Name hinterher nicht mehr gefällt, oder Sie wollen eine Datei auf eine andere Diskette kopieren, auf der eine Datei gleichen Namens bereits besteht. In all diesen Fällen können Sie das eingebaute Kommando REN oder RENAME bei 3.0 (zu deutsch: Umbenennen) benutzen, um Ihre Dateien umzutaufen. Das allgemeine Format für den RENAME-Befehl lautet:

RENAME Neu=Alt

Zuerst kommt also der neue Name der Datei, dann der alte. Zwischen den beiden Namen steht ein Gleichheitszeichen. Wenn die "Umtaufe" auf dem angemeldeten Laufwerk gemacht werden soll, brauchen Sie keine Laufwerksbezeichnung vor die Dateinamen zu schreiben. Möchten Sie einer Datei einen Namen geben, der bereits existiert, werden Sie von CP/M 2.2 gewarnt:

File exists

Das neue CP/M ist da ein wenig hilfreicher und Sie werden gefragt, ob die alte Datei gelöscht werden soll.

Eine einfache Umtaufe einer Datei sieht so aus:

REN neu.bas=alt.bas

Sie können auch bei RENAME 3.0 wieder das Sternchen und die Fragezeichen benutzen. Ein einfaches Beispiel könnte so aussehen

RENAME *.TXT=*.BAK

Auf diesen Befehl hin werden alle Back-up-Dateien dieser Diskette in Dateien mit der Kennung TXT umgewandelt, ganz gleich, was in den Dateien drinsteht. Der Befehl RENAME kümmert sich nämlich überhaupt nicht um den Inhalt von Dateien, sondern nur um die Namensgebung.

IV.7 TYPE

Mit diesem Befehl lassen Sie sich Dateien auf dem Bildschirm anzeigen. Das funktioniert allerdings nur mit Text-Dateien, weil diese mit den Zeichen des ASCII-Zeichensatzes gebildet werden. Andere Dateien, mit der Kennung COM, REL oder ähnlich, enthalten Steuerzeichen, die jede Bildschirmausgabe durcheinander bringen und den Computer veranlassen, sich "aufzuhängen". Den TYPE-Befehl geben Sie so ein

TYPE A:DATEI.XXX

Wenn sie eine Datei auf einem anderen Laufwerk sehen wollen, stellen Sie eine andere Laufwerksbezeichnung davor.

Zu diesem Befehl gibt es bei CP/M 3.0 eine Option: NO PAGE. Damit lassen Sie den angezeigten Text kontinuierlich über den Bildschirm rollen. Normalerweise stoppt die Bildschirmausgabe nach jeweils 23 Zeilen und geht erst nach einem ENTER weiter.

Wenn Sie NO PAGE mit eingegeben haben, können Sie mit Control S (^S) die Bildschirmausgabe anhalten und mit Control Q (^Q) weiterlaufen lassen. Haben Sie vorher noch ein Control P (^P) eingetippt, wird die Bildschirmausgabe auf Ihrem Drucker aufgelistet.

IV.8 Was Sie jetzt schon wissen

* *CP/M besitzt sowohl eingebaute Befehle, wie auch solche, die erst durch ein Programm auf der Diskette ermöglicht werden*

* *Befehle können zusammen mit Parametern oder Optionen eingegeben werden*

* *Sie wissen, wie alle eingebauten Befehle heißen, wie sie wirken und welche Optionen nur mit den transienten Befehlen möglich sind*

V. Transiente Befehle

Von den eingebauten Befehlen haben Sie jetzt schon so einiges gelesen. Die wirkliche Macht des Betriebssystems liegt aber in den transienten Befehlen, die als COM-Datei im Inhaltsverzeichnis zu finden sind. Der erste Teile dieses Kapitels beschäftigt sich mit Befehlen des CP/M 2.2, und da besonders mit STAT.

Im zweiten Teil, der erheblich mehr Umfang hat, lernen Sie die vielfältigen Möglichkeiten von CP/M 3.0 kennen, die Ihnen mit Sicherheit den Mund wässrig machen werden.

V.1 Statusanzeigen mit STAT

Der Befehl STAT kann zum Einen zur Anzeige des Systemstatus, zum Anderen für die Änderung der Gerätezuordnung in CP/M genutzt werden. Die einfachste Form des Befehls, die sicherlich auch sehr oft gebraucht wird ist diese:

STAT

mit einem anschließenden ENTER. Damit können Sie sich den noch verfügbaren Speicherplatz auf der Diskette anzeigen lassen. Das sieht so aus:

A:R/W,SPACE:89K

Diese Angabe bedeutet, daß Sie noch 89 Kilobyte Platz auf der Diskette haben, die gelesen und beschrieben werden kann.

Letzteres ersehen Sie aus der Angabe R/W, was für Read/Write steht und Lesen/Schreiben heißt. Im Gegensatz dazu gibt es auch den Diskettenstatus R/O (Read only; nur lesen), wenn die Diskette mit einem Schreibschutz versehen ist. Dann können Sie auf diese Diskette nichts schreiben, sondern nur Vorhandenes lesen.

Genauso interessant wie der verbleibende Platz auf einer Diskette, kann aber auch die Größe einer Datei oder eines bestimmten Typs von Dateien sein. Sie können sich die Größe einer Datei auf einem beliebigen Laufwerk ansehen, wenn Sie den entsprechenden Laufwerksnamen mit eingeben:

STAT B:DATEN.CMD

Sie erhalten dann diese Auskunft über den Bildschirm:

```
    RECS BYTES EXT Acc
    ------------------------------
      12 2K 1 R/W B:DATEN.CMD

Bytes Remaing on A: xK
```

Benutzen Sie bei dieser Abfrage das Sternchen als Joker, etwas so:

STAT B:*.CMD

bekommen Sie entsprechend viele Anzeigen im obigen Format.

Das Feld RECS gibt an, wieviele Sätze die jeweilige Datei belegt. In dem Feld BYTES steht die Größe einer Datei in Einheiten von 1K, während unter EXT die Zahl der zusammenhängenden Blöcke abzulesen ist.

Möchten Sie die gesamte Diskette mit einem Schreibschutz versehen, können Sie mit STAT das R/O-Attribut setzen. R/O ist das Gegenteil des Attributs R/W und gestatt nur, Dateien zu lesen. Um solch einen Schreibschutz zu verwirklichen, geben Sie ein:

STAT B: $R/O

Genauso können Sie vorgehen, wenn Sie statt einer Diskette nur eine einzelne Datei oder ein Gruppe von Dateien schützen wollen:

STAT DATEI.BAS $R/O

Da die Zuordnung verschiedener USER-Bereiche bei CP/M 2.2 leider noch nicht im Prompt angezeigt wird, müssen Sie den STAT-Befehl benutzen, um festzustellen, in welchem Benutzerbereich Sie sich befinden. Zur Belohnung für die Mühe bekommen Sie aber auch gleich angezeigt, welche Benutzerbereiche sonst noch Dateien enthalten und auf welchem Benutzerbereich diese stehen. Sie geben also ein:

STAT USR:

und bekommen diese Übersicht:

Active User: 0

Active Files:0 2 3 5 11 7

Daraus ersehen Sie, in welchem Benutzerbereich Dateien stehen, denn wenn nicht mindestens eine Datei in einem Bereich steht, wird auch bei dieser Übersicht nichts angezeigt.

V.2 Transiente Befehle bei CP/M 3.0

Jetzt kümmern wir uns um wichtige und häufig gebrauchte CP/M 3.0-Befehle. Es handelt sich um transiente Befehle, die Sie alle in Ihrem Inhaltsverzeichnis finden, mit einem COM als Kennung versehen. Sie wissen ja bereits, wie Sie Programme aufrufen: den Namen der Programme ohne die Kennung eintippen und ein ENTER danach geben. Sie haben drei Möglichkeiten die transienten Programme aufzurufen. Einmal, wenn Sie im USER-Bereich 0 sind und dort auch arbeiten wollen, zum Zweiten, wenn Sie die Programme mit dem SET-Befehl für alle USER-Bereiche zugänglich gemacht haben, drittens können Sie noch den Befehl SETDEF benutzen. Über den letztgenannten Befehl erzähle ich Ihnen später noch mehr.

V.3 SET

Der SET-Befehl übernimmt verschiedene Aufgaben bei CP/M 3.0. Zur Hauptsache wird er wohl verwendet, um verschiedene Datei-Attribute zu setzen. Das sind Zusätze zu den Dateien, die bestimmte Wirkungen haben. Zum Beispiel benutzen Sie den SET-Befehl, um Ihre CP/M-Dateien im USER-Bereich 0 für alle USER-Bereiche zugänglich zu machen. Weiterhin können Sie aber auch Ihre Dateien schützen, indem Sie sagen: "Diese Datei darf nur gelesen werden" oder "Diese Datei ist durch ein Geheimwort geschützt".

Der SET-Befehl beinhaltet aber auch Optionen, die das Inhaltsverzeichnis beeinflußen. Zum Beispiel können Sie sich ein Inhaltsverzeichnis erstellen lassen, in dem jede Datei einen "Zeit-Stempel" enthält, der darüber Auskunft gibt, wann sie erstmals angelegt und wann zum letzten Mal damit gearbeitet wurde.

Bevor Sie ein Inhaltsverzeichnis mit den "Timestamps", den "Zeit-Stempeln" einrichten können, müssen Sie das Programm INITDIR.COM laufen lassen. Glauben Sie nicht, die Stempelei sei nur ein Spaß. Sie können diese Einrichtung später nutzen, um alle Dateien die verändert wurden, automatisch auf eine andere Diskette zu sichern.

Damit Sie die vielfältigen Möglichkeiten von SET näher kennenlernen, hier erst einmal eine Übersicht über die verschiedenen Möglichkeiten

```
OPTION BEDEUTUNG
====== =========

DIR Macht eine SYSTEM-Datei wieder im
                          normalen Inhaltsverzeichnis sichtbar
```

SYS Macht eine Datei zur SYSTEM-Datei

RO Bestimmt, daß eine Datei nur gelesen werden kann

RW Bestimmt, daß eine Datei gelesen und verändert werden kann

ARCHIV=OFF Setzt das ARCHIV-Attribut auf "aus". Das bedeutet, daß diese Datei noch nicht gesichert (archiviert) wurde. Das Programm PIP mit der Option ÄAÜ kann Dateien mit dem Attribut ARCHIV=OFF kopieren. Den PIP-Befehl geben Sie mit den Sternchen für die Dateinamen ein und PIP kopiert alle Datei, die seit dem letzten Kopieren mit PIP und der ÄAÜ-Option verändert wurden. Nachdem PIP kopiert hat, setzt es die Datei-Attribute auf ARCHIV=ON.

ARCHIV=ON Setzt das ARCHIV-Attribut auf "ein". Das bedeutet, daß diese Datei gesichert wurde. Normalerweise ändert PIP mit der Option ÄAÜ das Attribut nach dem Sichern von Dateien. Sie können das Attribut auch selber ändern, wenn Sie den SET-Befehl verwenden.

F1=ON/OFF Schaltet das benutzerdefinierte Datei-Attribut F1 ein oder aus.

F2=ON/OFF Schaltet das benutzerdefinierte Datei-Attribut F2 ein oder aus.

```
F3=ON/OFF Schaltet das benutzerdefinierte Datei-
                    Attribut F3 ein oder aus.

F4=ON/OFF Schaltet das benutzerdefinierte Datei-
                    Attribut F4 ein oder aus.
```

Nun haben Sie gesehen, was es alles gibt und sollen jetzt die Anwendung kennenlernen. Sie haben einige wertvolle Programme, zum Beispiel Ihr Textprogramm oder die Datenbank und wollen damit aus allen USER-Bereicher heraus arbeiten. Normalerweise stehen diese Programme, genau wie die CP/M-Programme, im Benutzer-Bereich 0. Damit Sie auf diese Programme oder auch andere Dateien immer zugreifen können, verwenden Sie den SET-Befehl mit einer oder mehreren Optionen und machen die entsprechende Datei zu einer SYSTEM-Datei. Das sieht so aus:

SET PIP.COM[SYS]

Wollen Sie sichergehen, daß in die Datei PIP.COM (oder jede andere) nichts hineingeschrieben wird, geben Sie eine zweite Option mit an:

SET PIP.COM[SYS RO]

Damit ist PIP.COM jetzt eine System-Datei und zudem schreibgeschützt. Wollen Sie eine so gesicherte Datei wieder frei zugänglich machen und das SYSTEM-Attribut aufheben, geben Sie ein:

SET PIP.COM[DIR RW]

Dabei ist es gleichgültig, in welcher Reihenfolge Sie die Optionen eingeben.

V.4 Laufwerk-Attribute

Sie haben auch die Möglichkeit, ganze Laufwerke so zu definieren, daß von ihnen nur gelesen, aber nicht darauf geschrieben werden kann. Haben Sie ein Laufwerk auf "RO", was für "read only" (nur lesen) steht gesetzt, kann keine Datei mit ERASE gelöscht werden, RENAME funktioniert nicht und PIP kann keine Dateien darauf kopieren.

Sobald Sie ein Control C eingeben (Control-Taste drücken und festhalten und dann zusätzlich noch "C" drücken) ist der RO-Status des Laufwerks wieder aufgehoben. Wenn Sie:

SET A:[RO]

eingeben, ist das Laufwerk A: schreibgeschützt. Geben Sie RW, (heißt read/write = lesen/schreiben) statt RO ein, kann das Laufwerk wieder voll genutzt werden.

V.5 Labels

Besonders wenn Sie viele Disketten haben oder mehrere Benutzer an einem Computer arbeiten, ist die Möglichkiet nützlich, den Disketten Namen geben zu können. Dazu benutzen Sie wieder den SET-Befehl, aber mit der Option "NAME=". Die Disketten-Namen unterliegen den gleichen Beschränkungen, wie die Datei-Namen. Das heißt, Sie können maximal acht Zeichen für den Namen verwenden und drei Zeichen für die Kennung.

Um eine Diskette mit einem Namen zu versehen geben Sie ein:

SET B:[NAME=FIBU]

Das wäre eine Bezeichnung zum Beispiel für eine Diskette mit Ihrer Finanzbuchhaltung drauf. Wenn Sie schon auf dem Laufwerk B: arbeiten, können Sie die Laufwerksbezeichnung auch weglassen. Sie sehen den Namen einer Diskette nicht in Ihrem Inhaltsverzeichnis, wenn Sie es mit DIR aufrufen. Dazu benutzen Sie den Befehl SHOW, Näheres wird später erklärt, und dazu die Option "LABEL". Die Option können Sie auf ein einfaches "L" abkürzen und der komplette Befehl sieht so aus:

SHOW B:[L]

wobei Sie die Laufwerksbezeichnung wieder vergessen können, wenn Sie sowieso schon auf B: arbeiten.

V.6 PASSWORD

Damit Ihnen niemand Ihre schöne Disketten-Ordnung durcheinander bringt, gibt es in CP/M die Möglichkeit ein Kennwort zu vereinbaren. Sie schützen also den Namen Ihrer Diskette mit einem Kennwort, das nur Sie kennen.

Sobald Sie ein Kennwort für eine Diskette oder eine einzelne Datei eingegeben haben, verlangt CP/M vor jeder Befehlsausführung genau dieses Kennwort. Deshalb legen Sie sich besser eine Datei mit allen Kennworten an, damit Sie selber auf Dauer zurechtkommen. Haben Sie erst einmal ein Kennwort vergessen, gibt es für Sie keine Möglichkeit, mehr an Ihre Daten heranzukommen. Also - Vorsicht.

Um also Ihr Label mit einem Kennwort zu versehen, benutzen Sie den SET-Befehl mit der Option "PASSWORD=Kennwort" oder im umgekehrten Fall "PASSOWRD=<cr>" (<cr> steht für RETURN). Das Kennwort geben Sie so ein:

SET [PASSOWRD=GEHEIM]

oder, um ein Kennwort aufzuheben:

SET [PASSWORD=<cr> (<cr> = ENTER)

Nochmal: Denken Sie daran, Ihr Kennwort irgendwo aufzubewahren. Wenn Sie es nicht mehr wissen, haben Sie keinen Zugriff auf Ihre Daten auf dieser Diskette.

V.7 PROTECT

Sie können nicht nur das Label Ihrer Diskette mit einem Kennwort versehen, sondern auch einzelne Dateien oder sogar CP/M-Befehle. So können Sie sicherstellen, daß wirklich kein Unbefugter an Ihren Daten herumspielt oder Einblick in Dinge bekommt, die ihn nichts angehen.

Bevor Sie nun eine einzelne Datei oder einen Befehl schützen können, müssen Sie den PROTECT-Modus einschalten. Das geschieht einfach dadurch, daß Sie folgenden Befehl eingeben:

SET [PROTECT=ON]

oder umgekehrt:

SET [PROTECT=OFF]

Wenn Sie das eingegeben haben sind Sie bereit, Ihre Dateien zu schützen.

V.8 Datei-PASSWORD

Um eine Datei unter Ihren persönlichen Schutz zu stellen, gehen Sie im Prinzip genauso vor, wie vorher beim Schützen von Disketten-Namen.

Sie können auch mehrere Dateien gleichzeitig unter den Schutz eines Kennwortes stellen, wenn Sie die Möglichkeit der Sternchen (*) benutzen. Dann gilt für alle Dateien, auf welche die Bedingung zutrifft, das gleiche Kennwort. Die Befehlseingabe sieht so aus:

SET TEXT.TXT[PASSWORD=Kennwort]

oder mit "*" formuliert:

SET *.TXT[PASSWORD=Kennwort]

In dem ersten hier gezeigten Beispiel ist die Datei TEXT.TXT durch "Kennwort" geschützt, im zweiten Beispiel bekommen alle TXT-Dateien das Kennwort "Kennwort".
Zusätzlich läßt sich noch festlegen, wie die Dateien geschützt werden sollen. Dazu gibt es folgende Möglichkeiten:

READ Das Kennwort wird benötigt, um Dateien zu
lesen, kopieren, beschreiben, löschen oder
umzubenennen

WRITE Das Kennwort ist nötig, um Dateien zu

beschreiben, löschen oder um einen neuen
Namen zu geben

DELETE Das Kennwort ist nur zum Löschen notwendig

NONE Es gibt kein Kennwort. Wenn bereits ein
Kennwort vorhanden ist, kann es mit dieser
Option gelöscht werden

Eingegeben wird das so:

SET TEXT.TXT[PROTECT=DELETE]

Damit haben Sie die Datei TEXT.TXT vor versehentlichem Löschen geschützt.

V.9 TIME STAMP

Mit der Einrichtung des Zeit-Stempels, können Sie sich einen Überblick darüber verschaffen, wie oft Sie eine bestimmte Datei benutzen, wann Sie eine Datei erstellt und geändert haben und Ihre Sicherungs-Dateien managen. Ein Zeit-Stempel ist wie die Stechuhr in einer Firma, womit alle Arbeitszeit-Daten festgehalten werden.

Um Zeit und Datum zu jeder Datei festhalten zu können, müssen Sie zuerst das Programm INITDIR laufen lassen. Das Verwalten der Zeiten ist nur möglich, wenn das Inhaltsverzeichnis in einer besonderen Weise organisiert ist. Das erreichen Sie mit INITDIR.

Sie haben drei verschiedene Optionen zur Verfügung

```
CREATE=ON Schaltet die Zeit-und Datummarkierung
                    für die Dateien auf einem bestimmten
                    Laufwerk ein.

ACCESS=ON Schaltet die Markierung des letzten
                    Zugriffs auf eine Datei ein. Sie können
                    aber nur CREATE oder ACCESS einschalten.
                    Wenn ACCESS auf einem Laufwerk gewünscht
                    wird, für das bereits CREATE gewählt
                    wurde, wird CREATE automatisch
                    ausgeschaltet.

UPDATE=ON Sorgt für eine Markierung der Dateien,
                    jedesmal wenn sie neu bearbeitet wurden.
                    Das Anzeigen in einem Inhaltsverzeichnis
                    hat keine Wirkung auf diesen Zeit-
                    Stempel.
```

Wenn Sie UPDATE und CREATE als Ihre Optionen wählen, sollten Sie beachten, daß bei einer Bearbeitung der Dateien beide Zeit-Stempel aktualisiert werden.

Das liegt daran, daß beim Bearbeiten einer Datei eine neue eingerichtet wird, während die alte zur BAK-Datei (Sicherheits-Kopie) wird.

Ihre Computer-Stechuhr bringen Sie so zum Laufen:

SET [ACCESS=ON]

Um sich das Ergebnis dieser Operation anzusehen, gibt man den Befehl:

DIR[FULL]

und bekommt dies:

```
Directory for Drive B:

Name Bytes Recs Attributes Prot Update Access
------- ----- ---- ---------- ---- ------------- -------------
TEXT.TXT 5K 38 DIR RW NONE 04/01/85 17:31
FIBU 20K 152 SYS RO NONE 04/01/85 09:10
```

Unter der Überschrift ACCESS können Sie ablesen, wann Sie welche Datei zuletzt benutzt haben. Der Befehl für zwei Eintragungen in das Inhaltsverzeichnis sieht so aus:

SET [CREATE=ON,UPDATE=ON]

Ein Inhaltsverzeichnis hätte dann diesen Aufbau:

```
Directory for Drive B:

Name Bytes Recs Attributes Prot Update Create
------- ----- ---- ---------- ---- ------------- -------------
```

```
TEXT.TXT 5K 38 DIR RW NONE 04/17/85 10:00 01/01/85 09:00
FIBU 20K 152 SYS RO NONE 04/17/85 16:43 01/01/82 19:21
```

V.10 SETDEF

Dieser Befehl bietet Ihnen die Möglichkeit, nach Programmen auf Ihren Disketten suchen zu lassen. Sie arbeiten beispielsweise auf dem Laufwerk B:, haben aber alle CP/M-Dateien oder andere Programme auf dem Laufwerk A: gespeichert. Mit SETDEF teilen Sie CP/M nun einen Suchpfad mit, damit es die gewünschten Programme lädt, ohne daß Sie jedesmal die Laufwerksbezeichnung mit eingeben müssen.

Wenn Sie einfach:

SETDEF

eingeben, erhalten Sie Informationen darüber, wie Ihr Suchpfad derzeit aussieht, welches Laufwerk für temporäre Dateien verwendet wird und nach welchen Datei-Typem gesucht wird. Geben Sie:

SETDEF A:

ein, sucht CP/M immer auf dem A:-Laufwerk nach den gewünschten Dateien, selbst wenn Sie auf Laufwerk B: arbeiten. Wenn Sie den Befehl so erweitern:

SETDEF A:,*

sucht CP/M zuerst auf dem A:-Laufwerk und dann auf dem angemeldeten nach den Dateien. Der Stern steht für das gerade angemeldete Laufwerk.

Sollen temporäre Dateien, also Zwischendateien, wie PIP zum Beispiel welche erstellt, auf ein besonderes Laufwerk, geben Sie

SETDEF [TEMPORARY=C:]

ein und die Zwischendateien werden auf ein drittes Laufwerk oder auch in die RAM-Disk geschrieben.

Gesucht werden kann nur nach Dateien, die eine COM oder SUB Kennung haben. Normaleinstellung bei CP/M 3.0 ist die Suche nach COM-Dateien, was aber so:

SETDEF [ORDER=(SUB,COM)]

geändert werden kann. SUB-Dateien sind solche, die mit dem Befehl SUBMIT aufgerufen werden und hintereinander ausführbare Befehle enthalten. Dazu kommen wir gleich noch.

V.11 SHOW

Wir kommen jetzt zu einem Befehl, der Ihnen viele Informationen über den Platz auf Ihren Disketten, die Namen Ihrer Disketten und über die Anzahl der Dateien pro USER geben kann. Wenn Sie einfach:

SHOW

eintippen, sehen Sie die Attribute aller Laufwerke und den noch verbleibenden Platz auf jedem Laufwerk.

A:RW, Space: 101K

B:RW, Space:5,934K

Geben Sie SHOW mit einer Laufwerksbezeichnung dahinter an, bekommen Sie nur die statistischen Angaben zu diesem einen Laufwerk.

Mit SHOW können Sie sich aber auch, wie schon weiter oben erwähnt, die Labels Ihrer Disketten anzeigen lassen:

SHOW A:[LABEL]

wobei Sie LABEL auch auf ein schlichtes "L" abkürzen dürfen. Auf dem Bildschirm sehen Sie dann:

```
Label for drive A:

Directory Passwds Stamp Stamp
Label Reqd Create Update Label Created Label Updated
------------ ------- ------ ------ -------------- --------------
FIBU .COM off off off 04/17/85 11:41 04/17/85 11:41
```

Mit einer weiteren Option zum SHOW-Befehl können Sie sich anzeigen lassen, welche USER-Bereiche auf Ihrer Diskette genutzt werden und wieviele Dateien zu jedem Bereich gehören. Zusätzlich bekommen Sie noch die Anzahl der freien Inhaltsverzeichnis-Einträge angezeigt:

A: Active User : 0
A: Active Files: 0 2 11 12
A: # of files : 85 6 1 1

A: Number of free directory entries: 24
Wenn Sie einfach:

SHOW A:[DIR]

eingeben, bekommen Sie nur die Zahl der noch freien Einträge angezeigt.

V.12 SUBMIT

Bisher haben Sie alle Befehle schön brav über die Tastatur eingegeben. Das ist ja alles ganz logisch und auch gut so, aber wenn Sie immer wieder die gleichen Befehle eingeben sollen, um immer das Gleiche zu erreichen, wird's auf die Dauer doch lästig. CP/M kann Sie von dieser Last befreien.

Mit dem Befehl SUBMIT können Sie eine ganze Reihe von Befehlen abarbeiten lassen, die in einer Datei stehen und wie Eingaben über die Tastatur verarbeitet werden. Eine Datei, in der solche Befehlsfolgen stehen, wird mit der Kennung SUB versehen und kann dann von dem Programm SUBMIT gelesen und ausgeführt werden.

Wenn Ihr Computer zum Beispiel über keine eingebaute und durch eine Batterie am Laufen gehaltene Uhr verfügt, müssen Sie die Zeit und das Datum jedesmal nach dem Start des Computers neu eingeben. Wollen Sie sich selber zwingen, diesen Eintrag jedesmal vorzunehmen, schon damit die Zeit-Stempel richtig und kontinuierlich gesetzt werden, verwenden Sie die Datei PROFILE.SUB.

Diese Datei hat eine besondere Eigenschaft: Sie wird bei jedem Start von CP/M 3.0 gelesen und die darin enthaltenen Befehle ausgeführt. Wenn Sie hier also vorgeben, daß Zeit und Datum eingetragen werden sollen, kommen Sie um die Eingabe nicht herum. Die Datei schreiben Sie sich so:

B:DATE SET

mit Ihrem Textprozessor und taufen diese Datei PROFILE.SUB. Natürlich können Sie auch ein anderes Laufwerk angeben. Das hängt alleine davon ab, auf welchem Laufwerk die Datei

DATE.COM zu finden ist. Vergessen Sie die Kennung SUB, führt der SUBMIT-Befehl gar nichts aus.
Eine SUB-Datei kann CP/M-Befehle, verschachtelte SUBMIT-Befehle und Eingabedaten für ein Programm oder einen CP/M-Befehl enthalten. Wenn Sie etwas darüber nachdenken werden Sie feststellen, daß damit eine ganze Menge zu machen ist.

Sie können in einer SUB-Datei auch mit allgemeinen Anweisungen arbeiten, die entsprechend Ihren Eingaben dann verwendet werden. Diese allgemeinen Anweisungen heißen Parameter und werden durch ein Dollarzeichen ($) dargestellt. Verwenden können Sie Parameter von $1 bis $9.

Nehmen wir an, Sie schreiben sich eine Datei, die so aussieht:

ERA $1.BAK
DIR *.$2

und von mir aus DIR.SUB heißen soll. Mit dieser Befehlsfolge in der Datei löschen Sie zuerst alle Datei eines bestimmten Namens und mit der Kennung .BAK. Dann lassen Sie sich alle Dateien mit einer bestimmten Kennung zeigen. Um dorthin zu gelangen geben Sie ein:

SUBMIT TEXT COM

und löschen zuerst alle Dateien, mit dem Namen TEXT.BAK und bekommen anschließend eine Übersicht über alle Dateien mit der Kennung COM. Die übersetzte SUBMIT-Datei sieht so aus:

ERA TEXT.BAK
DIR *.COM

und CP/M führt diese Befehle genauso aus, als wären sie über die Tastatur eingegeben worden.

Geben Sie weniger Parameter ein, als in der SUBMIT-Datei vorgesehen sind, werden die übrigen Parameter nicht beachtet. Geben Sie mehr Parameter ein, als in der SUBMIT-Datei stehen, werden die überzähligen ebenfalls nicht beachtet. Möchten Sie ein Dollarzeichen in einem Befehl innerhalb einer SUBMIT-Datei stehen haben, geben Sie einfach zwei Dollarzeichen hintereinander ein ($$) und Sie bekommen, was Sie wünschen.

Eine SUBMIT-Datei kann aber nicht nur einzeilige Befehle enthalten, sondern auch Befehlseingaben für Programme. Sie können also mit einer Befehlszeile ein Programm aufrufen und mit der nächsten Befehle an das aufgerufene Programm weitergeben. Ein Beispiel:

PIP
<B:=A:*.COM
<
DIR *.COM

Hier sehen Sie eine kleine SUBMIT-Datei, die etwas Neues enthält. Mit der ersten Zeile rufen Sie PIP.COM auf, geben in der zweiten Zeile den Befehl, alle COM-Dateien auf Laufwerk B: zu kopieren, steigen dann aus PIP aus und sehen sich anschließend das Inhaltsverzeichnis an. Alle Befehle, die direkt an ein Programm gehen, werden mit dem "Kleiner als"- Zeichen (<) kenntlich gemacht. Geben Sie es ohne

irgendeinen Zusatz ein, wie in der dritten Zeile, bedeutet es ein RETURN, was wiederum, in diesem Fall, PIP beendet.

Der SUBMIT-Befehl ist aber nicht nur Spielerei, sondern kann durchaus nützlich angewendet werden. Sie können damit zum Beispiel beliebig viele Datein hintereinander drucken lassen und in der Zwischenzeit etwas essen gehen. Die Kommando-Datei schreiben Sie einfach so:

PIP LST:=DATEI1
PIP LST:=DATEI2
PIP LST:=DATEI3
PIP LST:=DATEI4
usw.

oder, mit noch weniger Aufwand kommen Sie so hin:

PIP LST:=DATEI?

Sie nennen die Datei dann SAMMELDR.SUB und geben, bevor Sie weggehen, ein:

SUBMIT SAMMELDR (RETURN)

Dabei können die Dateien natürlich auch auf verschieden Laufwerken stehen. Sie müssen nur den Laufwerksbuchstaben vor den Dateinamen schreiben und SUBMIT sucht sich den Rest zusammen.

Sie könnten sich auch eine Kommando-Datei schreiben, die alle Dateien einer Festplatte über alle USER-Bereiche hinweg mit DIR auflistet, die Bildschirmausgabe dann aber in eine neue Datei mit dem Namen Inhalt schreibt. In dieser Datei suchen Sie eine bestimmte Datei dann mit dem Suchbefehl Ihres Textprogramms oder drucken die Dateien-Liste aus.

V.13 Der HELP-Befehl

CP/M 3.0 ist mit seinen vielen Befehlen nicht mehr so leicht zu erlernen und beherrschen wie seine Vorgänger. Besonders die Optionen werden Sie sicherlich nicht so schnell alle im Kopf behalten. Glücklicherweise bietet das Betriebssystem seine Hilfe sozusagen "auf Knopfdruck" an. Sie müssen nur auf englisch "Hilfe" schreien und schon kommt, was Sie brauchen. Mit dem Schreien war natürlich das Eintippen in den Computer gemeint. Sie geben also einfach:

HELP

ein und bekommen ein Menü mit den möglichen Hilfestellungen angeboten. Da suchen Sie sich einen Unterpunkt aus und werden ausführlicher informiert.

```
HELP UTILITY V1.1

At "HELP>" enter topic [,subtopic]...

EXAMPLE: HELP> DIR EXAMPLES

Topics available:
```

```
COMMANDS CNTRLCHARS COPYSYS DATE DEVICE DIR

DUMP ED ERASE FILESPEC GENCOM GET

HELP HEXCOM INITDIR LIB LINK MAC

PATCH PIP (COPY) PUT RENAME RMAC SAVE

SET SETDEF SHOW SID SUBMIT TYPE

USER XREF



HELP>
```

Sie können die Unterpunkte auch direkt aufrufen, indem Sie zum Beispiel eingeben:

HELP SETDEF

damit kommen Sie direkt an die Hilfsinformationen heran. Doch leider ist nicht alles Gold was glänzt und englisch nicht jedermannes Muttersprache. Selbstverständlich gehen unsere Freunde die amerikanischen Programmierer davon aus, daß die Welt englisch spricht. Nach langem Hin und Her und nach vielen durchkauten Kaugummis haben sich die CP/M-Erfinder aber durchgerungen, eine Möglichkeit für Nicht-Amerikaner zu schaffen, die Hilfsinformationen auch in anderen Sprachen auf den Schirm zu holen.

Das Hilfe-Programm besteht aus den Dateien HELP.COM und HELP.HLP. Um hier etwas zu ändern, rufen Sie die Datei HELP.COM auf und geben Sie als Option EXTRACT ein:

HELP [EXTRACT]

Sie können das EXTRACT auch auf ein kurzes "E" abkürzen. Das HELP-Programm erstellt dann aus der Datei HELP.HLP eine neue Datei mit dem Namen HELP.DAT, die Sie mit Ihrem Textverarbeitungssystem nach Ihren Vorstellungen und in Ihrer Sprache umändern können.

Um neue Hilfstexte einzugeben, müssen Sie folgendes beachten:

Jedes Stichwort muß mit drei Schrägstrichen (///) und einer Nummer beginnen. Die Nummer gibt die Hilfsstufe des Stichwortes an. Zum Beispiel:

///1DIR (ENTER)

///2OPTIONEN (ENTER)

///3PARAMETER (ENTER)

///4BEISPIELE (ENTER)

Haben Sie alles geändert oder vielleicht eine Hilfe für ein wenig benutztes Programm installiert, speichern Sie die Datei ab und rufen wieder HELP.COM auf, allerdings diesmal mit der Option CREATE, abgekürzt "C". Daraufhin wird eine neue HELP.HLP-Datei erstellt, die Ihre Änderungen enthält.

V.14 Was Sie jetzt schon wissen

* *Sie haben den gewichtigen Befehl STAT von CP/M 2.2 kennegelernt und können sich damit die verschiedenen Informationen über Ihr System holen oder diese ändern*

* *Sie haben die transienten Befehle von CP/M 3.0 kennengelernt*

* *Sie kennen mittlerweile eine große Anzahl von Optionen, mit denen Sie die Wirkung der transienten Befehle verändern oder erweitern können*

* *Sie wissen, daß Sie Dateien mit einem Label versehen können*

* *Sie haben gesehen, wie man eine ganze Diskette, eine Datei oder bestimmte CP/M-Befehle vor unbefugtem Zugriff schützt*

* *Ihnen ist auch klar, wie Sie Ihre Dateien mit einem Stempel für die Zeit und das Datum präparieren können*

* *Sie haben die Möglichkeiten des Suchpfades für Programme kennengelernt*

* *Sie wissen nun, wie Sie die System-Daten einer Diskette mit SHOW anzeigen lassen können*

* *Die Datei PROFILE.SUB können Sie verwenden, um die Startroutine Ihres Computers zu automatisieren*

VI. Alles über PIP

Von dem Wunderding namens PIP haben Sie jetzt schon einige Male gehört. Damit Sie nicht glauben, ich übertreibe hier schamlos, zähle ich Ihnen schnell auf, was PIP alles kann

* *Eine einzelne Datei von einer Diskette auf eine andere übertragen*

* *Eine Gruppe von Dateien von einer Diskette auf eine andere übertragen*

* *Eine Datei kopieren und mit einem anderen Namen versehen*

* *Text zum Ausdrucken formatieren*

* *Zu lange Zeile kürzen*

* *Eine Sammlung von Datei auf einen Befehl hin drucken*

* *Mehrere Dateien zu einer zusammenkopieren*

* *Ein Stück aus einer Textdatei herausholen*

* *Kleinbuchstaben in große verwandeln und umgekehrt*

* *Das achte oder Parity-Bit wieder auf Null setzen*

* *Eine Datei mit Zeilennummern versehen*

* *Eine Datei während des Übertragens auf dem Bildschirm anzeigen*

* *Systemdateien übertragen*

* *Dateien von einem USER-Bereich zu einem anderen kopieren*

* *Neu erstellte oder geänderte Dateien automatisch sichern*

Na, ist das genug? Auf jeden Fall Grund genug, sich mit diesem Programm ausführlich zu beschäftigen. Lesen Sie dieses Kapitel und das Handbuch lieber drei als zweimal und machen Sie sich klar, was PIP alles für Sie erledigen kann.

VI.1 Diskette kopieren

Nahezu jedes Handbuch zu irgendeinem Programm beginnt mit dem Rat, die Originaldiskette erst einmal mit PIP auf eine neue, formatierte Diskette zu überspielen und dann nur noch diese Kopie zu benutzen. In einem früheren Kapitel hab' ich Sie auch darauf hingewiesen, und gezeigt wie es geht. Weil die Denkweise von PIP etwas verschieden von der normalen Denkweise einfacher Menschen ist, hier noch mal das einfache Beispiel. Sie kopieren von einer Diskette auf eine andere so:

PIP
***B:TEXT.TXT=A:TEXT.TXT**

Damit erreichen Sie, daß die Datei TEXT.TXT von der Diskette im Laufwerk A: unter dem gleichen Namen auf dem Laufwerk B: nochmal erzeugt wird. Ihr Original bleibt wie es ist, Sie haben nur eine Kopie auf einem anderen Laufwerk.

Um eine komplette Diskette zu kopieren, geben Sie ein

PIP
B:=A:*.

damit schaufeln Sie alle Dateien von A: nach B:, allerdings ohne das Betriebssystem zu kopieren. Das müssen Sie mit einem gesonderten Befehl auf die Systemspuren jeder Diskette schreiben. Dieser Befehl heißt:

SYSGEN bei CP/M 2.2 und
COPYSYS im CP/M 3.0 (unter DISCKIT3!!)

Haben Sie beides kopiert, besitzen Sie eine voll verwendbare, das heißt auch "bootbare", Kopie Ihres Originals. Sie können die Kopie in's Laufwerk schieben und den Computer starten. Um die oben vorgestellte Prozedur zu vereinfachen und zu beschleunigen gibt's zwei Möglichkeiten.

Der Befehl PIP kann auch mit einer ganzen Reihe von Optionen versehen werden, die die unterschiedlichsten Auswirkungen haben. Eine Option heißt "V" und steht für "verify", was übersetzt "nachprüfen" bedeutet. Sobald PIP mit dieser Option arbeitet, prüft es nach jedem Kopiervorgang genau nach, ob die neue Datei auch wirklich genau der Ursprungsdatei entspricht. Die Optionen müssen wieder mit eckigen Klammern eingegeben werden. Um die Datei TEXT.TXT zu übertragen und gleichzeitig zu überprüfen geben Sie ein:

PIP
***B:=A:TEXT.TXT [V]**

Das war der erste Schritt. Allerdings finde ich es lästig, jedesmal PIP aufzurufen und dann erst in der zweiten Zeile den Befehl zu geben. Aus CP/M-Sicht ist das auch gar nicht erforderlich. Sie können den oben beschriebenen Befehl auch so abkürzen

PIP B:=A:TEXT.TXT [V]

Dann beginnt PIP sofort mit der Arbeit und kehrt danach direkt zum Prompt (A>) zurück. Sicher haben Sie gemerkt, daß ich für's Laufwerk B: keinen Dateinamen eingegeben habe. Wenn der Name der Datei beim Kopieren nicht verändert werden soll, können Sie genauso vorgehen. Möchten Sie den Namen aber ändern, geben Sie ein:

PIP B:TEXT1.TXT=A:TEXT.TXT

Damit können Sie Dateien gleichzeitig kopieren und den Namen ändern.

Bevor Sie aber mit PIP auf eine andere Diskette kopieren, sollten Sie sich vergewissern, ob dort auch genug Platz für die neue Datei ist. PIP überträgt nämlich die Datei auf die andere Diskette, errichtet dort eine Zwischendatei, die Sie am gleichen Dateinamen, aber mit einer $$$-Kennung, erkennen können. Erst wenn feststeht, daß die Datenübertragung erfolgreich war, benennt PIP die Zwischendatei auf den richtigen Namen um.

Nehmen wir an, Sie kopieren die Datei TEXT.TXT auf die Diskette B:, aber dort steht noch , mit gleichem Namen, die alte Version dieser Datei. Was macht PIP? Wenn die Diskette nicht mehr genug Platz bietet, bekommen Sie eine Fehlermeldung. Das liegt an der Arbeitsweise von PIP. Es

versucht zuerst, die zu kopierende Datei auf die Diskette zu schreiben, obwohl die alte Version dort auch noch steht. Daher der Platzbedarf. Ist genügend Platz vorhanden, wird die neue Datei übertragen, mit $$$ gekennzeichnet, dann erst die alte Datei gelöscht und die neue umbenannt. Klappt also Ihre Kopier-Operation einmal nicht, haben Sie möglicherweise nicht genug Platz für die neue Datei und müssen erst mit ERASE die alte Datei löschen.

Bei jeder Übertragung durch PIP ohne spezielle Option, werden die Datei-Attribute wie SYS,DIR,RO und RW mit übertragen. Wenn Sie also eine SYSTEM-Datei mit PIP kopieren, ist auch die Kopie eine SYSTEM-Datei.

Verlangen Sie von PIP, eine Datei zu übertragen und auf der Zieldiskette existiert bereits eine Datei gleichen Namens, aber schreibgeschützt (RO=Read Only), fragt PIP bei Ihnen nach, ob es diese Datei löschen darf.

Antworten Sie auf die Frage mit "Y" für Ja und mit "N" für nein.

VI.2 Zwischen Benutzer-Bereichen kopieren

Was Sie bis jetzt gelesen haben, ermöglichte nur die Übertragung von Dateien innerhalb eines USER-Bereiches. Befinden Sie sich im USER-Bereich 3 und verwenden PIP zum Kopieren einer oder mehrere Dateien, landen die Kopien auf der anderen Diskette auch im USER-Bereich 3. Denken Sie daran, wenn Sie Ihre übertragenen Dateien einmal vermissen sollten.

Um nun von einer Diskette und gleichzeitig von einem USER-Bereich in einen anderen zu kopieren, benutzen Sie die

Option "Gn", wobei das "n" für die Nummer des USER-Bereiches steht. Um vom Benutzer-Bereich 0 die Datei TEXT.TXT auf Laufwerk B: in den Benutzer-Bereich 2 zu kopieren, geben Sie ein:

PIP B:[G2]=A:TEXT.TXT

Denken Sie daran, daß andere CP/M-Befehle wie DIR,ERASE,TYPE usw. nur in dem jeweils angewählten USER-Bereich etwas ausführen.

VI.3 Text- und Nicht-Textdateien

Bis jetzt haben Sie PIP immer nur benutzt, um Dateien von einer Diskette auf die andere zu bringen. Wie ich oben schon erwähnte, haben Sie auch die Möglichkeit, aus mehreren Dateien eine Gesamtdatei zu erstellen, und zwar in einem Arbeitsgang.

Bevor Sie über dieses interessante Feature mehr erfahren, muß ich Ihnen noch schnell etwas zu den unterschiedlichen Datei-Formen sagen. Grundsätzlich unterscheidet CP/M zwei Dateiformen: Die Text-Datei und die Nicht-Text-Datei. Eine Textdatei erstellen Sie mit Ihrem Textverarbeitungs-Programm oder dem in CP/M eingebauten Editor ED (lieber nicht) und verwenden dabei alle Zeichen, die Ihre Tastatur so hergibt. Eine Nicht-Text-Datei besteht aus Binär-Code und trägt in der Regel die Kennung COM.

Diese beiden Dateiarten müssen unterschieden werden, weil CP/M das Ende von Text-Dateien an einem Control Z (^Z) erkennt. Jeder Text, der nach dem ^Z geschrieben steht, wird nicht ausgedruckt oder sonstwie beachtet. In Nicht-Text-Dateien dagegen, ist ein ^Z ein ganz normales Zeichen, wie viele andere auch und CP/M schert sich nicht darum.

Normalerweise geht PIP immer davon aus, daß Nicht-Text-Dateien zu übertragen sind und liegt damit häufig richtig. Wenn Sie aber mit PIP mehrere Dateien zu einer zusammenfassen, geht PIP davon aus, daß es sich um Text-Dateien handelt.

VI.4 Dateien zusammenkopieren

Als Erkennungszeichen für PIP, daß es mehrere Dateien zusammenlegen soll, dient das Komma. Aus diesem kühlen Grunde sollen Sie auch kein Komma in einem Dateinamen benutzen. Sie bringen sonst PIP völlig durcheinander.

Eine einfache Zusammenlegung verschiedener Dateien veranlassen Sie so:

PIP ALLES.TXT=TEIL1.TXT,TEIL2.TXT,TEIL3.TXT

wenn alle Dateien auf dem gleichen Laufwerk stehen. Soll die Sammel-Datei in einen anderen USER-Bereich geschrieben werden, geben Sie bei CP/M 3.0 ein:

PIP ALLES.TXT[G3]=TEIL1.TXT,TEIL2.TXT,TEIL3.TXT

Bestehen Sie zusätzlich noch auf der Überprüfung nach dem Kopieren, müssen Sie hinter jede Datei, die Teil der neuen werden soll ein [V] eingeben. Damit sieht die Befehlszeile dann schon etwas komplizierter aus:

PIPALLES.TXT[G3]=TEIL1.TXT[V],TEIL2.TXT[V],TEIL3.TXT[V]

Auf jeden Fall ist das immer noch weniger Arbeit, als alle drei Datei "von Hand" zusammenzutragen.

Wollen Sie Nicht-Text-Dateien mit PIP zusammenfassen, kann es zu Problemen kommen. Wie Sie gelesen haben, denkt PIP jedesmal wenn ein ^Z auftaucht, die Datei sei zu Ende. In Nicht-Text-Dateien kann das ^Z häufiger vorkommen und würde so eine Übertragung mit PIP unmöglich machen oder doch sehr erschweren.

Weil die CP/M-Erfinder dieses Problem kennen und davon ausgehen, daß meistens COM-Dateien als Nicht-Text-Dateien übertragen werden, übersieht PIP gnädig jedes ^Z in einer COM-Datei und kopiert bis zum bitteren Ende derselben.

Wenn Sie Dateien kopieren möchten, die weder Text-Dateien noch COM-Dateien sind, müssen Sie den Parameter "O" mit angeben. Wie gehabt, mit den eckigen Klammern direkt hinter dem Dateinamen.

Sie haben sogar die Möglichkeit, während der Zusammenstellung mehrerer Dateien in jede Datei noch etwas hineinzuschreiben. Der Kopierbefehl muß dann so aussehen:

PIP ALLES.TXT=TEIL1.TXT,CON:,TEIL2.TXT,CON:,TEIL3.TXT

So beginnt PIP die erste Datei zu übertragen, stoppt dann und erwartet Ihre Eingaben über die Tastatur, angeregt durch den Befehl CON:.

Allerdings müssen Sie die Befehl für ENTER und "neue Zeile" (^J) von Hand eingeben, damit PIP keine Zeile überschreibt. Mit der Eingabe von ^Z geht's dann weiter. Bedenken Sie aber, daß Schreibfehler nicht korrigiert werden können.

VI.5 Zeilen durchnummerieren

Eine nette Einrichtung für Programmierer, Journalisten und alle, die ihre Texte oder Programme zeilenweise nummeriert haben möchten ist der "N"-Parameter von PIP. Wenn Sie eine Datei mit diesem Parameter oder seinem Bruder "N2" kopieren, bekommt jede Zeile Ihres Textes eine eigene Nummer.

Kopieren Sie so:

PIP TEXTNR.TXT=TEXT.TXT[N]

erscheinen Ihre Textzeilen hinterher so:

1: Dies ist Ihr Text

Nehmen Sie dagegen den Parameter "N2", erscheint Ihr Text hinterher so:

000001 Die ist Ihr Text

Die Zeilen werden einfach hochgezählt, was sich nicht ändern läßt. Also in der Reihenfolge: 1,2,3,4,5...

VI.6 Buchstaben umwandeln

Möchten Sie statt Zeilenummern lieber Ihren gesamten Text in Großbuchstaben erscheinen lassen, sind Sie ein Kandidat für den Parameter "U". Das "U" steht für "Uppercase", was Großbuchstaben bedeutet. Das Gegenteil bekommen Sie, wenn Sie den Parameter "L" für "Lowercase" eingeben. Dann besteht Ihr gesamter Text nur noch aus kleinen Buchstaben.

VI.7 String suchen

Manchmal passiert es, daß der Drucker streikt oder einfach mitten im Druck das Papier zu Ende geht. Dann haben Sie unter Umständen das Problem, daß ein Teil Ihres Ausdruckes zwar noch auf der Diskette steht, aber nicht auf dem Papier. Bevor Sie nun die ganzen 200 Seiten nochmal ausdrucken, benutzen Sie lieber einen weiteren PIP-Parameter, um den restlichen Text auf's Papier zu bekommen.

Sie können mit PIP einen Textteil kopieren, indem Sie PIP mitteilen, ab welchem String, das ist eine Zeichenkette, es suchen und bei welchem String es wieder aufhören soll. Den Anfangs-String markieren Sie durch ein "S" für "Start" und den End-String mit einem "Q" für "Quit", was aufhören bedeutet. Schreiben Sie hinter jede Zeichenkette ein ^Z und PIP macht den Rest. Es durchsucht Ihren Text bis es auf die vorgegebene Zeichenkette stößt, kopiert und stoppt bei der zweiten, von Ihnen vorgegebenen Zeichenkette. Ein solcher Befehl sieht so aus:

PIP
***TEXTTEIL=TEXT.TXT[Qsuchwort^Z]**

Wenn Sie das so schreiben, beginnt PIP am Anfang der Datei mit dem Kopieren und stoppt, sobald das Wort "suchwort" auftaucht. Beachten Sie bitte, daß PIP ohne Option aufgerufen und erst in der zweiten Zeile alles andere eingegeben wurde. So sucht PIP nach dem "suchwort", wie Sie es eingetippt haben, nämlich in Kleinbuchstaben.

Schreiben Sie den obigen Befehl in einer Zeile, wandelt PIP das "suchwort" erst in Großbuchstaben um und beginnt dann mit der Suche. Steht Ihr "suchwort" nicht als "SUCHWORT" in Ihrer Datei, findet PIP nichts und gibt Ihnen eine Fehlermeldung aus.

Einen Textabschnitt kopieren Sie mit diesem Befehl:

PIP
***TEXTTEIL=TEXT.TXT[Sanfang^ZQende^Z]**

Das Programm beginnt mit dem Kopieren beim Wörtchen "anfang" und hört beim ersten Auftauchen des Wortes "ende" auf.

VI.8 Mehrere Dateien hintereinader drucken

Sie kennen bisher die Methode eine Datei auszudrucken, indem Sie ^P eingeben und dann mit dem Befehl TYPE die Bildschirmausgabe auf den Drucker umleiten. Das ist etwas

umständlich und kann mit Hilfe von PIP besser gelöst werden. Jedes Peripherie-Gerät hat für PIP einen eigenen Namen. Das Ausgabegerät wird mit LST: bezeichnet. Die Doppelpunkte sind wichtig, damit PIP diesen Gerätenamen von einem Dateinamen unterscheiden kann. Wollen Sie Ihre Textdatei auf dem Drucker erscheinen lassen, geben Sie ein:

PIP LST:=TEXT.TXT

Sie können selbstverständlich alle Optionen, die PIP beeinflussen, auch bei dieser Operation verwenden. Um zum Beispiel mehrere Dateien hintereinander zu drucken, geben Sie ein:

PIP LST:=TEXT1.TXT,TEXT2.TXT,TEXT3.TXT

und der Text kommt hintereinander aus Ihrem Drucker gequollen. Wenn Sie die ausgedruckte Form Ihres Textes beeinflussen wollen, benutzen Sie den Namen PRN:. Sie bekommen dann durchnummerierte Textzeilen, acht Zeichen breite Tabulator-Schritte und alle 60 Zeilen wird ein neues Blatt begonnen.

Am Ende dieses Kapitels gebe ich Ihnen einige Beispiele, was man mit PIP Sinnvolles machen kann.

VI.9 Dateien automatisch sichern

Nun ist es aber nicht nur sehr sinnvoll, Daten auszudrucken, sondern genauso wichtig, Daten zu sichern. Ich habe Sie

bereits darauf hingewiesen und spreche aus leidvoller Erfahrung. Bei meinen Wanderungen durch das PIP-Handbuch habe ich dann eine PIP-Funktion entdeckt, die das Sichern wesentlich erleichtert. Es ist die "Archiv"-Option, abgekürzt "A".

Besonders wenn Sie mit einer Harddisk arbeiten, werden Sie die Vorteile dieses Parameters schnell kennenleren. Jede Datei in CP/M 3.0 besitzt im Dateikopf einen Platz für ein Bit, das mit "archive flag" bezeichnet wird. Ein Flag ist sowas wie ein Anzeiger, der einen bestimmten Zustand anzeigt. Jedesmal, wenn Sie eine Datei eröffnen oder ändern, wird dieses Flag verändert. Also beim Ändern einer Datei zum Beispiel auf "1" gesetzt und nach erfolgter Datensicherung wieder auf "0".

So kann PIP erkennen, welche Dateien seit der letzten Sicherung verändert oder neu erstellt wurden und nur diese auf eine Diskette sichern. Sie ersparen sich damit, jedesmal die gesamte Harddisk zu sichern, was eine Menge Zeit und Geld für zusätzliche Disketten kosten würde.

Nach einem langen Arbeitstag sichern Sie Ihre bearbeiteten Dateien, indem Sie eingeben:

PIP B:=A:*.TXT[A]

Wollen Sie sicher gehen, setzen Sie noch die "V"-Option dazu und jede Datei wird auf vollständige Übertragung überprüft

PIP B:=A:*.TXT[AV]

Wenn Sie die Originale der gesicherten Dateien auf der Festplatte jetzt nicht mehr anrühren, werden diese beim nächsten Sichern nicht beachtet.

Sie können sich im Inhaltsverzeichnis ansehen, welche Dateien gesichert sind und welche nicht. Dazu benutzen Sie den Befehl DIR mit den Optionen FULL oder RW. Im Inhaltsverzeichnis sehen Sie die gesicherten Datein mit "Arcv" gekennzeichnet.

VI.10 Überschreiben ohne Rückfrage

Es gibt noch ein paar Feinheiten zu beachten, wenn Sie mit PIP arbeiten. Normalerweise überschreibt PIP beim Kopieren ohne zu zögern eine Datei auf der Zieldiskette, wenn sie den gleichen Namen besitzt. Handelt es sich dabei allerdings um eine Datei, die durch das Attribut R/O (Read Only) geschützt ist, fragt PIP extra nochmal nach, ob diese Datei überschrieben werden darf.

DESTINATION FILE IST R/O, DELETE (Y/N)?

Geben Sie hier "Y" für Ja ein, wird die Datei auf der Zieldiskette überschrieben und damit gelöscht. Bestehen Sie auf Ihrem Nein, indem Sie "N" drücken, bricht PIP die Kopieroperation ab. Nach "Y" oder "N" brauchen Sie übrigens kein RETURN zu drücken.
Möchten Sie, daß auf jeden Fall alle Dateien auf die Zieldiskette geschrieben werden und die Frage nach der Löscherlaubnis unterbleibt, verwenden Sie den "W"-Parameter.

Der Befehl sieht dann so aus:

PIP B:=A:TEXT1.TXT,TEXT2.TXT,TEXT3.TXT[W]

Damit übernehmen Sie aber auch die volle Verantwortung für's Kopieren und können PIP keinen Vorwurf machen.

VI.11 System-Dateien kopieren

Versuchen Sie eine Datei zu kopieren, die das SYSTEM-Attribut besitzt, bringen Sie PIP in Verlegenheit. Es kann sie nämlich nicht finden. Sie müssen ihm schon ein wenig helfen und die "R"-Option mit auf den Weg geben. Natürlich können Sie die beiden Optionen auch kombinieren. Sollen Dateien und SYSTEM-Dateien auf die Zieldiskette geschrieben werden, ohne Rücksicht darauf, ob sie schreibgeschützt sind oder nicht, geben Sie ein:

PIP B:=A:*.COM[RW]

Damit kopieren Sie alle COM-Dateien und überschreiben gleichnamige auf der Zieldiskette.

VI.12 Das 8. Bit "säubern"

Der amerikanische Zeichensatz (ASCII) wird mit Hilfe von sieben Bits dargestellt. Das achte Bit, das ist die im Computer verwendete Wortlänge, bleibt für Sonderaufgaben frei. WordStar zum Beispiel verwendet es, um die Zwischenräume beim Schreiben in Blocksatz zu markieren. Andererseits verlangt MBASIC, daß das achte Bit frei sein muß. Wenn Sie mit WordStar ein BASIC-Programm bearbeiten und fälschlicher-

weise den Dokumenten-Modus (D) gewählt haben, kann Ihr BASIC-Programm nicht laufen, weil das 8. Bit nicht frei ist. Dieses Problem bekommen Sie schnell in den Griff, wenn Sie die "Z"-Option von PIP benutzen. Sie kopieren einfach Ihre Datei mit PIP und alles ist wieder in Butter. Den Befehl dazu schreiben Sie so:

PIP SPIEL.BAS=SPIEL.BAS[Z]

Wenn Sie wollen, können Sie auch noch die "V"-Option dazu setzen. Aber - benutzen Sie den Befehl in dieser Form nicht, um WordStar-Dateien zu kopieren. Sie verlieren sonst die Formatierung des Textes.

VI.13 Praktische Beispiele

Sie sollen jetzt noch ein paar Beispiele sehen, wie Sie PIP sinnvoll einsetzten können. Die verschiedenen Parameter haben Sie in diesem Kapitel kennengelernt und wissen auch, daß man sie zusammen anwenden kann.

Einen Textausdruck in besserer Form als nur mit ^P oder einfachem LST: ergibt diese Befehlsform:

PIP LST:TEXT.TXT[NT8P60]

Damit wird die Datei TEXT.TXT zum Drucker geschickt, alle Zeilen werden durchnummeriert, Tabulatoren sitzen in jeder achten Spalte und die Seitenlänge beträgt 60 Zeilen. Wenn Sie einmal zurückblättern werden Sie feststellen, daß dies genau die voreingestellten Werte von PRN: sind. Jetzt wissen Sie, was Sie an PRN: haben: Weniger Arbeit.

Möchten Sie beim obigen Beispiel noch zusätzlich dafür sorgen, daß alle Buchstaben als Kleinbuchstaben erscheinen, geben Sie ein:

PIP LST:TEXT.TXT[NT8P60L]

Sie fügen also nur die "L"-Option dazu und schon klappt's.

Um die Archiv-Arbeit zu rationalisieren, schreiben Sie sich in CP/M 3.0 eine SUBMIT-Datei folgenden Inhalts:

PIP A:=B:*.*[WAR]

und nennen diese ARCHIV.SUB. Die Optionen "WAR" bewirken, daß SYSTEM-Dateien mit kopiert, schreibgeschützte Dateien ohne Nachfrage überschrieben und nur solche Dateien kopiert werden, die noch nicht archiviert sind. Wenden Sie diesen Befehl regelmäßig an, kann Ihren Daten eigentlich nichts passieren. Der Aufruf erfogt einfach mit:

SUBMIT ARCHIV

und der Rest geht automatisch. Sie können den Befehl noch erweitern, wenn Sie möchten. Mit der Option "V" werden die Dateien auf richtige Übertragung überprüft und durch den Parameter "E" bekommen Sie alles Kopierte auf dem Bildschirm angezeigt. Das "E" sollten Sie aber nur bei Textdateien verwenden, weil Sie sonst Schwierigkeiten bekommen.

Weitere Möglichkeiten wären, den DIR und den SHOW-Befehl in die SUBMIT-Datei mitaufzunehmen. Dann sehen Sie welche und wieviele Dateien kopiert werden sollen und wieviel Platz noch auf der Zieldiskette vorhanden ist.

VI.14 Was Sie jetzt schon wissen

* *PIP ist eines der leistungsfähigsten Unter-Programme von CP/M*
* *Sie können einzelne Dateien auf eine andere Diskette übertragen*
* *Sie können ganze Disketten auf einmal kopieren*
* *Mit PIP machen Sie aus mehreren Dateien eine*
* *Sie können ein Stück aus einer Datei herausholen*
* *PIP nummeriert für Sie automatisch die Zeilen einer Text-Datei*
* *Mit PIP können Sie zuletzt veränderte Dateien automatisch sichern*
* *Sie können Großbuchstaben in Kleinbuchstaben verwandeln und umgekehrt*
* *Sie können zwischen den verschiedenen Benutzer-Bereichen kopieren*

VII. CP/M intern

VII.1 Erstellen einer Sicherheitsdiskette von CP/M

Bevor wir uns mit ganzer Kraft in CP/M stürzen, sollten Sie zunächst unbedingt eine Sicherheitskopie Ihrer CP/M-Diskette anfertigen. Wie leicht kann es passieren, daß man versehentlich ein File auf der Diskette löscht oder gar die gesamte Diskette auf unerklärliche Weise zerstört wird. (Und glauben Sie mir, das kommt öfter vor, als Ihnen und mir recht sein kann.)

Um zu verhindern, daß man dann wie ein Fragezeichen in der Landschaft nach einer lauffähigen CP/M-Diskette sucht, sollte man eine Kopie, auch Backup genannt, von der vorhandenen CP/M-Systemdiskette anfertigen.

Sie wissen mittlerweile, wie man eine Diskette formatiert, und wie man einzelne Files oder die ganze Diskette kopiert. Sollten Sie über nur ein Laufwerk verfügen, so bedienen Sie sich des Kommandos

DISCCOPY

Die Routine formatiert Ihnen wenn nötig die Zieldiskette, und teilt Ihnen auf dem Bildschirm mit, ob Sie die Quelldiskette (Source) oder die Zieldiskette (Destination) einlegen sollen.

Sie müssen die Disketten während des Kopiervorganges einige Male wechseln. Sollten Sie über zwei Diskettenlaufwerke verfügen, so empfiehlt sich das Kommando **COPYDISC**, das schneller, weil ohne Diskettenwechsel arbeitet.

Nachdem Sie ein oder zwei Sicherheitskopien von der CP/M-Diskette angefertigt haben, kann es losgehen.

VII.2 Diskettenformate der Schneider-Floppy

Die Schneider-Floppy verfügt über drei verschiedene Diskettenformate: Das Standard-CPC-Format, das Datenformat und das IBM-Format. Diese drei Formate haben einige Gemeinsamkeiten:

- *Es werden 40 Spuren formatiert (Spur 0 bis Spur 39).*
- *Die Sektorlänge beträgt 512 Bytes.*
- *Es passen 64 Einträge in das Disketteninhaltsverzeichnis.*

Standard-CPC-Format und Datenformat werden mit je neun Sektoren pro Spur formatiert. Diese beiden Formate unterscheiden sich lediglich darin, daß beim Datenformat die beiden oberen Spuren nicht mit CP/M belegt werden und somit der Programm- und Datenspeicherung zur Verfügung stehen. Das IBM-Format sticht insofern hervor, daß nur acht Sektoren pro Spur formatiert werden. Als Benutzer des Betriebssystemes CP/M kommt also das Datenformat für Sie nicht in Frage, da die wichtigen Spuren 0 und 1 nicht durch das CP/M belegt werden.

Wir wollen uns auf das Standard-CPC-Format beschränken, da das Datenformat nichts mit CP/M zu tun hat und das IBM-Format (und andere) später beschrieben wird. In diesem Format werden die beiden "oberen" Spuren 0 und 1 wie folgt belegt:

```
Track 0, Sektor &41         : Boot-Sektor.
Track 0, Sektor &42         : Konfigurationssektor.
Track 0, Sektor &43 bis &47: Nicht benutzt.
```

```
Track 0, Sektor &48, &49 und
Track 1, Sektor &41 bis &49: CCP und BDOS.
```

(& leitet Hexadezimalzahlen ein.)

CCP bedeutet Console Command Processor, BDOS steht für Basic Disc Operating System.
Sie wissen, daß Sie mit dem transienten Kommando FORMAT Ihre Disketten formatieren können. Beim CPC können Sie jedoch lediglich im Laufwerk A: Ihre Disketten formatieren. Da wir verschiedene Formate haben, müssen Sie auswählen, in welchem Format CP/M 2.2 Ihre Diskette formatieren soll. Es existieren folgende Möglichkeiten:

```
FORMAT          : Formatiert im Standard-CPC-Format
FORMAT I        : Formatiert im IBM-Format
FORMAT V        : Formatiert im Vendor-Format
FORMAT D        : Formatiert im Daten-Format.
```

Sollten Sie es mit einem anderen Parameter als (S),I,V oder D versuchen, so wird FORMAT diesen Parameter reklamieren. (S) bedeutet, daß Sie beim Formatieren in System-Format das "S" auch weglassen können.

VII.3 Fremde Disketten-Formate

Doch vielleicht ist es für Sie interessant, auch andere Formate lesen bzw. schreiben zu können. Optimal wäre natürlich die Möglichkeit, diese fremden Formate auch auf dem CPC zu formatieren. Wir haben eben gelernt, daß beim Schneider die Sektoren 512 Bytes umfassen. Da der Floppy-Controller frei programmierbar ist, kann man diesen Wert leicht verändern. Als Sektorgrößen sind 128, 256, 512, 1024 usw. Bytes denkbar. Wie man den Floppyc-Controller (PD 765A) im einzelnen programmieren kann, ist ausführlichst im Floppy-Buch zum CPC beschrieben.

Am Floppy-Controller können zwei Laufwerke angeschlossen werden, mittlerweile auch 5.25"-Laufwerke. Eigentlich hat es ohnehin nur für diejenigen Leser Zweck, ein anderes Format zu lesen, die über ein 5.25"-Laufwerk verfügen, da nur diese CP/M-Software und Daten anderer Rechner einladen können. (Schließlich ist der Schneider CPC der erste Rechner mit 3"-Laufwerken.) Aber auch die Leser, die "lediglich" über das 3"-Laufwerk verfügen, dürften erstaunt über die Möglichkeiten sein, die sich ihnen mit ihrem Floppy-Controller bieten.

Zunächst ist festzustellen, daß es hunderte von Formaten unter CP/M 2.2 für 5.25"-Disketten gibt. Wir stellen Ihnen hier eine Tabelle zur Verfügung, die die Firma Digital Research zusammengestellt hat. Die Tabelle gibt Auskunft über die verschiedenen Parameter der einzelnen CP/M-Formate, und versetzt uns in die Lage, diese anderen Formate auf dem CPC zu realisieren. Wir beschränken uns hier auf die Formate für einseitige 5.25"-Disketten, da 8"-Disketten wohl nicht mehr so zahlreich existieren.

D	B/S	BLS	Cap.	SPT	BSH	BLM	EXM	DSM	DRM	ALO/1	CKS	OFF	PSH	PHM	VR	Sec1	letz	Skew	Nr.
S	128	1	83	18	3	7	0	82	31	80/0	8	3	0	0	0	1	18	5	<1>
S	128	1	83	18	3	7	0	82	63	C0/0	16	3	0	0	0	1	18	4	<2>
S	256	1	92	20	3	7	0	91	63	C0/0	16	3	1	1	0	1	10	1	<3>
S	256	2	92	20	4	F	1	45	63	80/0	16	3	1	1	0	1	10	2	<4>
D	128	1	123	30	3	7	0	122	63	C0/0	16	2	0	0	0	1	30	1	<5>
D	256	1	144	32	3	7	0	142	63	C0/0	16	4	1	1	0	1	16	1	<6>
D	256	1	148	32	3	7	0	147	63	C0/0	16	3	1	1	0	1	16	1	<7>
D	256	1	152	32	3	7	0	151	63	C0/0	16	2	1	1	0	1	16	1	<8>
D	256	1	152	32	3	7	0	151	63	C0/0	16	2	1	1	0	1	16	2	<9>
D	256	1	157	34	3	7	0	156	63	C0/0	16	3	1	1	0	1	17	1	<10>
D	256	2	171	36	4	F	1	84	63	80/0	16	2	1	1	0	1	18	2	<11>
D	256	2	171	36	4	F	1	84	127	C0/0	32	2	1	1	0	0	17	1	<12>
D	512	1	153	32	3	7	0	155	63	C0/0	16	1	2	3	0	1	8	1	<13>
D	512	1	152	32	3	7	0	151	63	C0/0	16	2	2	3	0	1	8	1	<14>
D	512	1	171	36	3	7	0	170	63	C0/0	16	2	2	3	0	1	9	2	<15>
D	512	1	190	40	3	7	0	189	63	C0/0	16	2	2	3	0	1	10	1	<16>
D	512	1	195	40	3	7	0	194	63	F0/0	16	1	2	3	0	0	9	1	<17>
D	512	2	166	36	4	F	1	82	63	80/0	16	3	2	3	0	1	9	1	<18>
D	512	2	190	40	4	F	0	94	63	80/0	16	2	2	3	0	1	10	2	<19>
D	512	2	190	40	4	F	1	94	63	80/0	16	2	2	3	0	1	10	1	<20>
D	512	2	190	40	4	F	1	94	63	80/0	16	2	2	3	0	1	10	2	<21>
D	1024	1	185	40	3	7	0	184	63	C0/0	16	3	3	7	0	1	5	1	<22>
D	1024	2	160	40	4	F	1	78	63	80/0	16	3	3	7	0	1	5	1	<23>

(Quelle: C'T 6/85, Verlag Heinz Heise GmbH)

Eine auf den ersten Blick sicherlich undurchsichtige Tabelle; allerdings beinhaltet sie sehr wichtige Informationen, ohne die wir, wie wir noch feststellen werden, unser Vorhaben nicht verwirklichen könnten.

Wichtig für diejenigen, die das Format <21> interessiert: Nach dem Lesen müssen die Daten erst einmal komplementiert werden; dies kann durch XORen mit &FF geschehen.

Doch hier erst einmal die Liste der Hersteller, zur besseren Identifizierung der einzelnen Formate:

<01>: Xerox
<02>: Lifeboat R2, TRS-80, Mayon
<03>: Eurocom
<04>: Osborne SD
<05>: Superbrain
<06>: MC-CP/M Ecma-70, MC-CP/M+ #7
<07>: Eurocom II Format 2
<08>: Alphatronics DD, NEC PC 8001
<09>: Olympia ETX-II, Philips P-2000
<10>: Spectravideo
<11>: TRS-M4
<12>: Bondwell-12
<13>: IBM CP/M-86
<14>: ADPS
<15>: Dec VT-180
<16>: Rentiki
<17>: Kaypro II
<18>: Olympia Boss A
<19>: SEL Delsy 2000
<20>: Newbrain, Mayon
<21>: Superbrain, HKM-ZDOS
<22>: Osborne DD
<23>: BASF-7120

Die Abkürzungen in der ersten und zweiten Zeile haben im einzelnen folgende Bedeutung:

D=Density. In dieser Spalte finden Sie die Dichte des Formats, also: **S=Single Density** (einfache Dichte) und **Double Density** Doppelte Dichte). Es ist nur eine Frage der Zeit, bis man auch CP/M-Format mit vierfacher Dichte vorfindet; die entsprechenden Disketten mit der Aufschrift Quad-Density existieren bereits.

B/S bedeutet **Bytes pro Sektor.** Wir sprachen bereits an, daß der CPC 512 Bytes pro Sektor speichert. 128, 256 oder 1024 Bytes pro Sektor sind aber realisierbar.

BLS bezeichnet die **Blockgröße** (Block Size) in KBytes. Bei einem Format mit 128 Bytes pro Sektor und einer BLS von 1 KByte werden also 8 Sektoren zu einem Block zusammengefaßt. Beim CPC besteht ein Block aus 2 Sektoren, also aus 1 KByte.

Cap. steht, Sie können es sich bestimmt schon denken, für **Capacity=Kapazität.** Hier wird also die Speicherkapazität des entsprechenden Formates in KByte angegeben. Der CPC hat eine Kapazität von 169 KBytes.

SPT heißt **Sectors per Track.** Man darf das Wort Sektor jetzt aber nicht physikalisch auffassen, sondern muß es logisch verstehen. Da man zunächst Sektoren von lediglich 128 Bytes hatte, faßte man diese 128 Bytes zu einem Record zusammen. SPT gibt nun also an, wie viele von diesen Records, also 128-Byte-Einheiten, auf einer Spur Platz finden. Der CPC hat 9 Sektoren zu 512 Bytes, das macht 9*512=4608 Bytes pro Track. Dividieren wir diese Zahl 4608 durch 128, so erhalten wir unseren Fakor SPT: 4608/128=36.

BSH und **BLM** stehen für **Block-Shift** und **Block-Maske.** Diese beiden Angaben legen die Blockgröße fest. Der Controller benötigt diese Angaben.

EXM heißt **Extent-Maske** und legt die Anzahl der Einträge im Directory fest.

DSM legt die **größtmögliche Blocknummer** fest.

Unter **DRM** finden Sie die **maximale Anzahl** der möglichen **Einträge im Directory** (-1 !).

AL0/1 steht für die **Verzeichnisgröße**, binär kodiert. Beim CPC werden 2 Blocks für das Inhaltsverzeichnis frei gehalten.

CKS gibt an, aus wieviel **logischen Sektoren** (Records) das Inhaltsverzeichnis besteht. Beim CPC sind dies 16.

OFF steht für **Spuroffset.** Hier befindet sich die Zahl der reservierten Spuren für CP/M.

PSH und **PHM** geben Informationen für das **Blocking** und **Deblocking** von physikalischen Sektoren in logische Sektoren.

Sec1 beschreibt die **Nummer des ersten Sektors, letz** die Nummer des **letzten Sektors** auf einer Spur. Beim CPC ist der erste Sektor der Sektor Nr. 1, der letzte hat die Nummer 9.

Als letztes haben wir noch eine Angabe des Skew-Faktors **SKEW**. Nur für absolute Insider von Interesse.

Nachdem wir all diese Angaben haben, wollen wir sehen, welche Informationen man dem CPC, oder besser, unserem Controller PD 765A, mitteilen kann. Dazu sehen wir uns den System-Speicher der Floppy im CPC an:

Sie finden diese Angaben im System-RAM an Adresse &A890-&A8A5.

A890,A891	SPT	Records pro Track (36)
A892	BSH	Block Shift (3)
A893	BLM	Block Maske (7)
A894	EXM	Extend Maske (0)
A895,A896	DSM	Maximale Blocknummer (170)
A897,A898	DRM	Maximale Einträge im Directory-1 (63)
A899,A89A	AL0/1	Verzeichnisgröße (C000h) binär kodiert, entspricht 2 Blocks
A89B,A89C	CKS	Anzahl der im Verzeichnis zu prüfenden Einträge (&0010) 16 Einträge
A89D,A89E	OFF	Spuroffset (2) belegte Systemspuren
A89F	FSC	Erster Sektor jeder Spur (041h)
A8A0	PST	Physikalische Sektoren pro Track (9)
A8A1	GPS	Länge des GAP3 bei Sektor Read/Write (2Ah)
A8A2	GPT	Länge GAP3 bei Track formatieren (52h)
A8A3	FLB	Filler-Byte bei Track formatieren (E5h)
A8A4	BPS	Bytes/Sektor (2) entspricht 512 Bytes
A8A5	RPS	Anzahl Records/Sektor (4)

Die Werte in Klammern geben die Defaultwerte, also die voreingestellten Werte an.

Das Format <13>, das IBM-Format, ist das einzige, das Ihnen der CPC automatisch zur Verfügung stellt.

Eigentlich ist die Realisierung eines anderen Formates kein Problem: Sie sehen ja, daß man dem Controller wirklich alles vorgeben kann. Die einzige Schwierigkeit, die uns bleibt, ist die Wahl der Länge des GAP3. Das GAP3 ist eine Lücke, die bei den Disketten verwendet wird, um geringe Geschwindigkeitsschwankungen des Laufwerkes abzufangen, da das

Laufwerk beim Schreiben eines Sektors mit an Sicherheit grenzender Wahrscheinlichkeit nicht dieselbe Geschwindigkeit hat wie beim Formatieren dieses Sektors. Es besteht dann die Gefahr, daß man bei langsamerer Geschwindigkeit in den (physikalisch) nächsten Sektor hinein schreibt. Dabei dienen die GAP3-Lücken als Buffer. Je größer dieses GAP3 ist, desto weniger Sektoren passen auf eine Spur. Wählt man aber das GAP3 zu klein, so steigt die Gefahr, daß man sich in einen Folgesektor schreibt. Ein GAP3 müssen Sie sich als eine Aneinanderreihung von &4E-Codes vorstellen, jedes andere Zeichen wäre genauso denkbar.

Lassen Sie uns einmal versuchen, auf unserem Schneider das OSBORNE-Format <22> zu realisieren.

Folgende wichtige Fakten können wir unserer Tabelle entnehmen:

1024 Bytes pro Sektor, 5 Sektoren pro Spur, Sektornr. 1 bis Sektornr. 5. Diese und die anderen Informationen müssen wir also in den Systemspeicher des Rechners schreiben. Wichtig ist, und das hat bei der Entwicklung des Maschinenprogrammes einige Zeit gekostet, daß Sie bei vorgesehenen 16-Bit-Werten auch wirklich 16 Bit abspeichern. Andernfalls sonst verschieben sich nämlich alle folgenden Informationen im Systemspeicher des Floppy-Controllers und Sie erhalten mit Sicherheit nicht das gewünschte Ergebnis.

Nachdem Sie die Systemspeicheradressen &A890-&A8A5 versorgt haben, ist die Basis für ein anderes Format geschaffen. Wie Sie dem Maschinenprogramm auf den nun folgenden Seiten entnehmen können, kann man die vorhandenen Daten leicht mit einer Tabelle und dem LDIR-Kommando in den Systemspeicher kopieren. Sie können nach wie vor die Systemroutinen verwenden, sie verlaufen reibungslos, wie unser Beispielprogramm zeigt. Zuerst werden die 40 Spuren mit je 5 Sektoren

formatiert. Nachdem dies geschehen ist, werden alle Sektoren probeweise beschrieben. Dies ist ein Test, der zeigen soll, ob auch wirklich alle Sektoren verfügbar sind. Es könnte ja sein, daß GAP3 zu groß oder zu klein gewählt wurde oder irgend etwas anderes nicht so geklappt hat, wie es sollte (so war es in der Entwicklung natürlich auch). Wenn Sie beobachten, wie schnell die Floppy die 40 Spuren formatiert, und wie schnell diese Spuren auch alle wieder überschrieben werden, dann erkennt man wirklich die wahre Leistungsfähigkeit des Diskettenlaufwerkes.

Doch hier erst einmal das Assemblerprogramm zur Formatierung in OSBORNE-Format:

```
 10
 20 ;Formatiern in Osborne-Format, JS 9/6/85
 30 ;
 40 ;
 50        ORG  #7000 ;Startadresse
 60 ;
 70 ;Initialisiert FDC mit neuen Werten:
 80 ;
 90        LD   BC,22      ;22 WERTE
100        LD   HL,FDC     ;TABELLE DER NEUEN WERTE
110        LD   DE,#A890   ;FLOPPY-SYSTEMSPEICHER
120        LDIR
130 ;
140        LD   E,0        ;DRIVE
150        LD   D,0        ;TRACK
160        LD   B,40       ;40 TRACKS
170 L1:    PUSH DE         ;RETTE DE
180        PUSH BC
190        LD   HL,FTAB    ;TABELLE FÜR FORMATIERUNG
200        LD   A,D        ;TRACK=>D
210        LD   B,5        ;5 SEKTOREN
```

```
220 LO:    LD   (HL),A     ;SPEICHER TRACK
230        INC  HL
240        INC  HL
250        INC  HL
260        INC  HL         ;ZEIGER AUF NAECHSTEN SEKTOR
270        DJNZ LO         ;NÄCHSTER SEKTOR
280        LD   C,#1       ;ERSTER SEKTOR
290        LD   HL,FTAB    ;TABELLE ZUR FORMATIERUNG
300        RST  #18
310        DEFW FORMAT     ;FORMATIERE SPUR
320        POP  BC
330        POP  DE
340        INC  D          ;NAECHSTE SPUR
350        DJNZ L1
360 ;
370        JP   TESTE      ;ÜBERPRÜFE ALLE SEKTOREN
380 ;
390 ;
400 FORMAT:     DEFW #C652      ;ADRESSE #C652
410        DEFB 7          ;IM FLOPPY-ROM
420 ;
430 FTAB: EQU  $
440 ;
450        DEFB 0          ;TRACK
460        DEFB 0          ;KOPFADRESSE DES DRIVES
470        DEFB 1          ;SEKTOR
480        DEFB 3          ;3=1024 BYTES
490        DEFB 0,0,3,3    ;SEKTOR 3
500        DEFB 0,0,5,3    ;SEKTOR 5
510        DEFB 0,0,2,3    ;SEKTOR 2
520        DEFB 0,0,4,3    ;SEKTOR 4
530 ;
540 ;
550 ;TABELLE DER CONTROLLER-WERTE
560 ;
570 FDC:  DEFB 40,0,3,7,0,184,0,63,0,#C0,0,16
```

```
580       DEFB 0,3,0,1,5,42,50,#E5,3,8
590 ;
600 ;PROBEWEISES BESCHREIBEN ALLE SEKTOREN
610 ;
620       ORG #7100      ;NEUE EINSPRUNGADRESSE !!
630 ;
640 TESTE:CALL #BB06     ;WARTE TASTENDRUCK
650       LD   E,0       ;DRV
660       LD   D,0       ;TRACK
670 LL1:  LD   C,5       ;SEKTOR
680       LD   B,5       ;ZÄHLER
690       LD   HL,#5000  ;BUFFER, DER GESPEICHERT WIRD
700 LL2:  RST  #18
710       DEFW WRITE     ;SCHREIBE EINEN SEKTOR
720       JR   NC,FEHLER ;FEHLER AUFGETRETEN
730       DEC  C         ;NAECHSTER SEKTOR
740       DJNZ LL2
750       INC  D         ;NAECHSTE SPUR
760       LD   A,D
770       CP   40        ;ALLE 40 SPUREN??
780       JR   NZ,LL1    ;NOCH NICHT ALLE
790       RET            ;RUECKSPRUNG
800 ;
810 WRITE:DEFW #C64E
820       DEFB 7         ;ADRESSE #C64E IM FLOPPY-ROM
830 ;
840 FEHLER:    EQU  $    ;FEHLER DURCH PIEPSEN ANZEIGEN
850       LD   A,7       ;BEEP
860       CALL #BB5A     ;AUSGEBEN
870       RET            ;BRICH KONTROLLE AB
880 ;
890 ;
900       ORG  #7200     ;NEUE EINSPRUNGSADRESSE
910 ;
920       LD   E,0       ;DRIVE
930       LD   D,1       ;SPUR
```

```
940       LD   C,2       ;SEKTOR
950       LD   HL,#5000  ;BUFFER
960       RST  #18
970       DEFW READ      ;LIES SEKTOR
980       RET            ;ENDE DER PROZEDUR
990 ;
1000 READ:DEFW #C666
1010      DEFB 7         ;ADRESSE #C666 IM FLOPPY-ROM
1020 ;
```

Bitte haben Sie dafür Verständnis, daß wir an dieser Stelle keinen BASIC-LOADER abdrucken - es wäre sicherlich unsinnig. Denn wollten Sie ein anderes als beispielsweise das OSBORNE-Format realisieren, so wäre der gesamte BASIC-LOADER praktisch nutzlos. Die Leser unter Ihnen, die ein anderes Format auf Ihrem CPC einlesen wollen, müssen ohnehin über einige Maschinensprachenkenntnisse verfügen, da "Ihr" CP/M das neue Format nicht so einfach verarbeitet. Sie müssen sich unter Zuhilfenahme der BDOS-Routinen die Daten per Maschinenprogramm einladen. Kleiner Tip: Der (Z-80-) Assembler Macro-80 erstellt, wie der Assembler ASM auf Ihrer CP/M-Diskette, auf Wunsch COM-Files!

Bei entsprechender Änderung der Tabelle FDC (Zeile 570-580) ist jedes der abgedruckten Formate möglich. Achten Sie nur darauf, daß die Werte in der Reihenfolge des Systemspeichers abgelegt werden. Alle in der Tabelle vorkommenden, im Systemspeicher aber nicht auftretenden Werte können vernachlässigt werden.

Um sich einmal eine Vorstellung zu machen, wie schnell 1024 Bytes nachgeladen werden, probieren Sie einmal folgendes aus: Da alle Spuren nach dem Formatieren beschrieben werden, muß irgendein Speicherbereich zum Speichern herhalten. Im folgenden Beispiel wurde der Speicherbereich &5000-&5400

dafür genommen. Geben Sie folgende BASIC-Zeilen vor dem Formatieren ein:

```
DEFINT I
FOR I=&5000 to &5400
  POKE I,I AND &FF
NEXT
CALL &7000 (Zum Starten der Formatierung)
```

Damit ist jetzt der Speicherbereich &5000-&5400 mit Daten belegt worden. (Es wird von 0 bis 255 hochgezählt und dann wieder bei 0 begonnen.) Dieser Speicher ist nach dem Aufruf **CALL &7000** auch auf der Diskette (200 Mal) abgespeichert worden.

An Adresse &7200 befindet sich noch eine kleine Routine, um einen 1024 Byte großen Sektor einzulesen. Bevor Sie diese Routine aber aufrufen, sollten Sie den Speicher löschen, sonst können Sie ja nicht kontrollieren, ob der Speicher auch wirklich mit den Daten von der Floppy versorgt worden ist:

```
FOR I=&5000 TO &5400
  POKE I,0
NEXT
```

Nachdem Sie den Speicher gelöscht haben, können Sie einmal einen beliebigen Block laden - es steht ja überall dasselbe auf der Diskette:

CALL &7200

und blitzartig ist der Sektor eingelesen. Sie können sich davon überzeugen, indem Sie sich den Speicherbereich anzeigen lassen:

```
OR I=&5000 TO &5400
  PRINT PEEK(I);
NEXT I
```

Als Ergebnis sollten Sie erhalten:

1 2 3 4 5 6 7 8 9 10 11 12 13 ....

Dies dient nicht nur als anschauliches Beispiel der Geschwindigkeit, in der die Daten geschrieben werden, sondern auch als Probe, ob wirklich 1024 Bytes gespeichert und geladen werden.

Noch ein kleiner - unter Umständen aber sehr wichtiger - Tip: Da beim CPC standardmäßig nur 512 Bytes große Sektoren eingelesen werden, steht auch nur ein Buffer mit 512 Bytes zum Einlesen der Sektoren zur Verfügung. Da aber beim Einlesen von 1024 Bytes dieser Buffer praktisch "automatisch" auf 1024 Bytes erweitert wird, werden unter Umständen wichtige Daten oder Programmteile überschrieben. Dies stellt aber kein sehr großes Problem dar, da Sie den Sektorbuffer an eine beliebige Stelle verlegen können. Der Vektor für den Sektorbuffer steht an Adresse:

&BE62, &BE63

Sie erreichen dies beispielsweise, indem Sie folgende Zeilen zu dem Assembler-Programm ergänzen:

```
135     LD HL,SEKBUF
137     LD (#BE62),HL     ;Buffer umlegen
1030 SEKBUF:  DS    1024 ;1024 Bytes als Buffer
```

VII.4 Die CP/M-Diskette

Doch nun zurück zu dem Diskettenformat, das wir nicht erst umständlich simulieren müssen. Zum Schneider CPC 464 und CPC 664 wird eine Diskette mitgeliefert, auf der sich sowohl LOGO von Digital Research befindet als auch CP/M 2.2. Wir wollen zunächst einmal näher betrachten, was sich alles auf der CP/M-Diskette befindet - es tummeln sich dort nämlich nicht nur CP/M-Kommando-Dateien, auch wenn diese den weitaus größten Teil ausmachen.

```
A: MOVCPM   COM : PIP      COM : SUBMIT   COM : XSUB     COM
A: ED       COM : ASM      COM : DDT      COM : LOAD     COM
A: STAT     COM : DUMP     COM : DUMP     ASM : AMSDOS   COM
A: FILECOPY COM : SYSGEN   COM : BOOTGEN  COM : COPYDISC COM
A: CHKDISC  COM : DISCCOPY COM : DISCCHK  COM : SETUP    COM
A: FORMAT   COM : CSAVE    COM : CLOAD    COM : EX1      BAS
A: EX2      BAS : ROINTIME DEM
```

Auf der Diskette befinden sich beispielsweise DEMO-Dateien für BASIC unter AMSDOS, die wir hier getrost vernachlässigen wollen. Beschränken wir uns auf die CP/M-Kommandos, so stellen wir fest, daß sich neben der standardmäßigen CP/M-2.2-

Kommandos noch einige weitere COM-Dateien auf der Diskette befinden, wie etwa CLOAD.COM und CSAVE.COM. Diese beiden Kommando-Dateien dienen dazu, CP/M-Files von Diskette auf das Kassettenlaufwerk zu kopieren (CSAVE) bzw. Files von Kassette auf Diskette zu kopieren (CLOAD). Wollen Sie beispielsweise das File FILECOPY.COM auf Kassette abspeichern, so geben Sie lediglich folgendes Kommando in Ihren Rechner ein:

CSAVE FILECOPY.COM (RETURN)

Das ist alles. Auf Ihrem Bildschirm erscheint dann:

CSAVE V2.0

Press REC and PLAY then any key:
Saving FILECOPY.COM block x

Wenn Sie an Ihrem Kassettenlaufwerk die zwei entsprechenden Tasten betätigt haben, können Sie eine beliebige Taste drücken, und der Kopiervorgnag beginnt. Die Ausgabe "Block x" teilt mit, aus, welcher Block momentan auf Kassette gespeichert wird. Nachdem alles auf Kassette abgelegt worden ist, erscheint auf dem Monitor:

CSAVE V2.0 finished

Es kann sehr praktisch sein, bestimmte Dateien auf Kassette zu speichern, die Sie mittels dem CLOAD-Kommando wieder einlesen können, da Kassetten ein sehr sehr preiswertes Speichermedium sind, im Gegensatz zu den (noch) sehr teuren 3"-Disketten. Auf diese Weise können Sie bestimmte Files archivieren. Warum auch nicht? Auf Großcomputern macht man ja nichts anderes, da werden regelmäßig die Plattenstapel auf die preiswerteren Magnetbänder kopiert.

VII.5 Der mitgelieferte Assembler ASM

Auf der CP/M-Diskette, die jedem DDI-1-Laufwerk beiliegt und selbstverständlich auch dem CPC 664 und dem CPC 6128, befindet sich ein Assembler. Diesen Assembler können wir mit ASM aufrufen. Doch Stopp! Freuen Sie sich nicht zu früh. Wie Sie vielleicht wissen, besitzt der Schneider als CPU einen Z-80- Prozessor. Vielleicht haben Sie bereits in mühevoller, schweißtreibender Arbeit die Programmierung in Maschinensprache erlernt. Dieser Assembler ist aber kein Z-80-Assembler, sondern ein 8080-Assembler. Für diejenigen unter Ihnen, die es noch nicht so genau wissen: Der 8080 ist älter als der Z-80 und gewissermaßen sein Vater. Der Z-80 versteht zwar die Programme, die auch sein Vater versteht, er hat sogar einen weit größeren Sprachschatz als dieser, aber er hat einen anderen Mnemocode. Nein, wir haben uns nicht verschrieben. Mnemocode nennt man die Schreibweise, in der Maschinenprogrammierer Ihre Maschinenprogramme kodieren. Sie wissen sicherlich, daß Maschinenprogramme ausschließlich aus Zahlen bestehen. Beispielsweise versteht der Z-80-Prozessor das Kommando &41 und kann es ausführen. Für einen Menschen ist es allerdings sehr schwer, sich hierunter etwas vorzustellen. Dazu hat man die Mnemocodes entwickelt, die diese Maschinencodes zeitweise ersetzen sollen. Beim Z-80 sieht dieser Mnemocode für das Kommando &41 so aus:

LD B,C

Hierunter kann sich der Maschinenprogrammierer sehr wohl etwas vorstellen, und es erleichtert die Arbeit doch sehr, wenn man nicht immer in irgendwelchen Tabellen nachschlagen muß, was der Code &41 beispielsweise für Konsequenzen hat.

Der Assembler hat nun die Aufgabe, diese Mnemocodes, die der Mensch "so gut" versteht, in maschinenverständliche Codes zu übersetzen. Der Prozessor beispielsweise kann mit dem Mnemocode LD B,C nun überhaupt nichts anfangen. Hier tritt der Assembler als Dolmetscher oder Schnittstelle hervor, der die Menschensprache in Maschinensprache übersetzt. Der Assembler unterstützt den Menschen bei seiner Arbeit aber noch durch einige weitere Hilfsmittel, auf die ich später kurz eingehen will.

Sie wissen jetzt also, was ein Mnemocode ist, und daß der ältere 8080-Prozessor der Vater des Z-80 ist. Ferner ist Ihnen bekannt, daß der Z-80 mehr Befehle beherrscht als der 8080. Alle CP/M-Programme müssen aber in 8080-Code programmiert sein, weil es auch CP/M-Rechner gibt, die mit einem 8080 bestückt sind. Das bedeutet nichts anderes, als daß sich ein Z-80-Programmierer lediglich auf die Kommandos beschränken muß, die der 8080 auch verstehen kann (und dabei auf einige wirklich bemerkenswerte Kommandos des Z-80 zu verzichten hat) Das ist aber leider nicht alles, denn man hat auch die Mnemocodes bei der Weiterentwicklung des 8080 zum Z-80 geändert. Erinnern wir uns an unser Kommando $41. Es passiert in beiden Prozessoren bei der Erkennung und Ausführung dieses Kommandos dasselbe: Es wird der Inhalt des C-Registers in das B-Register kopiert. Der Mnemocode sieht beim Z-80 wie folgt aus:

LD B,C

Beim 8080 sieht dasselbe (!!) Kommando so aus:

MOV B,C

Sie erkennen sicherlich schon die Problematik: Ihnen nutzen Ihre Z-80-Kenntnisse wenig, wenn Sie einen 8080 programmieren wollen, selbst wenn Sie sich auf die Kommandos beschränken, die der 8080 auch verstehen kann. Also: Den ASM-Assembler können Sie nur mit Mnemocodes füttern, die ein 8080-Assembler auch verstehen kann.

Doch wollen wir mal diesen Assembler näher betrachten, der es uns ermöglicht, eigene COM-Files zu programmieren.

VII.6 Die Bedienung des Assemblers ASM

In der Programmierung der Maschinensprache tritt häufig das Problem auf, daß man sich auf bestimmte Speicherstellen beziehen muß, beispielsweise um Register zu laden oder abzulegen. Weiterhin kommt es auch sehr häufig vor, daß man an eine Speicheradresse springt etc. Stellen Sie sich einmal vor, Sie hätten ein Programm geschrieben, daß an der Speicheradresse $5000 beginnt, und wollen es nun an die Adresse $4000 verschieben. Sie müßten alle Sprungadressen außer den relativen Sprüngen ändern. Oder wenn Sie ein Kommando ins Programm einfügen, verschieben sich alle relativen Sprünge und alle folgenden direkten Sprünge. Spätestens dann tritt der Fall ein, daß der Programmierer durchdreht. Um diese Arbeiten zu erleichtern, hat man bei Assemblern die Möglichkeit eingeräumt, Marken (Labels) zu definieren, die während der Assemblierung (das ist die Übersetzung der Mnemocodes in Maschinensprache) in das Programm eingesetzt werden. Fortan ist es kein Problem mehr, ein Kommando an einer beliebigen Stelle einzufügen oder eine Routine oder gar das gesamte Programm zu verschieben - der Assembler übernimmt die unangenehmen Arbeiten schon für Sie.

Wie ein solches Assembler-Programm aussieht, wollen Sie wissen? Für Sie ist das gar kein Problem. Auf Ihrer CP/M-Diskette wurde freundlicherweise ein Assemblerprogramm abgespeichert, mit dem wir auch die Handhabung des Assemblers üben wollen. Sehen Sie sich das Inhaltsverzeichnis Ihrer CP/M-Diskette noch einmal genau an. Wir haben ein File mit Namen DUMP.COM; dies ist, wie wir bereits wissen, ein COM-File, das wir unter CP/M starten können, indem wir lediglich den Namen eintippen:

DUMP <Filename>

wäre die Syntax in diesem Fall. Dump liefert Ihnen einen Ausdruck der angegebenen Datei in HEX-Format (Hex-Dump). Probieren Sie es ruhig einmal aus:

DUMP FILECOPY.COM

Auf dem Bildschirm erscheint dann der Hex-Dump der COM-Datei Filecopy, das Programm gewissermaßen. Sie können die Ausgabe mit <Ctrl>/S anhalten und mit <Ctrl>/Q wieder mit der Ausgabe fortfahren. Mittels <Ctrl>/C können Sie die Ausgabe abbrechen.

DUMP.ASM ist nun das Maschinenprogramm von DUMP, so wie es vom Programmierer kodiert worden ist. Sie können sich dieses File einmal auf dem Bildschirm (oder Drucker) ausgeben lassen. Geben Sie hierzu ein:

TYPE DUMP.ASM

Sie erkennen im Vorspann des Programmes, daß die Firma DIGITAL RESEARCH die Urheberrechte für dieses Programm besitzt. Das ist aber nicht weiter verwunderlich, da genau diese Firma ja auch CP/M entwickelt hat. Sie sehen auch einen weiteren wesentlichen Unterschied zwischen Assembler und Maschinensprache: Sie können Kommentare einfügen. Diese Kommentare beginnen mit einem Semikolon, damit der Assembler weiß, daß er ab dieser Position nicht mehr übersetzen muß. Ansonsten würde der Assembler ja natürlich versuchen, diesen Kommentar zu verstehen, sprich zu übersetzen, und könnte nur verständnislos den Kopf schütteln und sagen: Hier stimmt aber etwas nicht!! Auch wenn Ihnen dieses Programm sehr lang vorkommen sollte, glauben Sie es mir, es ist verdammt kurz. Wenn Sie bedenken, daß Betriebsysteme auch in Assembler kodiert werden:

Es gibt Betriebsysteme, die beim Ausdruck 2000 Seiten und mehr lang sind. Auch "kleinere" Programme, wie beispielsweise die Textverarbeitung TEXTOMAT auf Ihrem Schneider, kommen noch leicht auf 400 bis 500 Seiten. Sie sehen, daß die Programmierung in Maschinensprache eine sehr aufwendige Sache ist. Doch wollen wir dieses Assemblerprogramm einmal assemblierenm, damit Sie in Zukunft den Assembler selbständig nutzen können. Achten Sie darauf, daß Ihre Arbeitsdiskette jetzt nicht schreibgeschützt ist, denn wir müssen auch auf die Diskette schreiben, auf der sich der Quellcode befindet. Sie haben sicherlich schon eine Sicherheitskopie von Ihrer CP/M-Diskette gemacht?! (Wenn nicht, dann sollten Sie dies schleunigst tun.) Wir assemblieren nun "unser" Maschinenprogramm:

ASM DUMP

Hier erkennen Sie schon ein wichtiges Merkmal des

Assemblers: Sie brauchen keine Extension einzugeben, d.h. Sie müssen nicht DUMP.ASM angeben, da der Assembler die Erweiterung .ASM selbständig anhängt. Auf Ihrem Bildschirm erscheint:

CP/M ASSEMBLER - VER 2.0
0257
002H USE FACTOR
END OF ASSEMBLY

A>

Schnell ist er, unser Assembler, nicht wahr? Er erstellt nun zwei Dateien:

1) Ein Protokoll unter dem Namen DUMP.PRN
2) Ein Hex-Dump unter dem Namen DUMP.HEX

Sehen Sie sich zunächst einmal das Protokoll an. Hier wird das Assembler-Programm noch einmal aufgelistet, mit allen Informationen, die der Assembler hinzufügt. In der ersten Spalte steht immer die Adresse, an der egrade kodiert wird - es sei denn, es wird ein Label definiert: Dann steht in der ersten Spalte der definierte Wert des Labels. Das Programm beginnt also bei Adresse $0100 und endet bei $0257; folglich wissen wir jetzt auch, was der Assembler uns da für eine Zahl ausgegeben hat (s.o.). Fast alle CP/M-Programme beginnen an Adresse $0100. Nachdem Sie sich das Protokoll gut angesehen oder es vielleicht sogar zu Papier gebracht haben, sehen Sie sich noch die zweite Datei an, die unser fleißiger Assembler erstellt hat. Dazu geben Sie wieder ein (Sie wissen es sicherlich mittlerweile):

TYPE DUMP.HEX

Sie sehen auf dem Bildschirm nun eine Reihe von Zahlen, die der Assembler erstellt hat; es handelt sich hierbei um die Programmcodes. Nebenbei werden noch einige weitere Informationen angezeigt, wie die Adresse, an der der Code später stehen soll etc. Diese Datei ist schon erheblich kürzer als unsere Datei DUMP.PRN, nicht wahr? Nur können wir dieses Maschinenprogramm immer noch nicht starten, es ist ja noch kein COM-File. Jetzt gibt es noch ein weiteres Programm, das diese Arbeit übernimmt. Es macht aus einer HEX-Datei eine COM-Datei, also ein von CP/M aufrufbares und startbares Programm. Dieses Programm trägt den Namen **LOAD**. Wir rufen dieses Programm einmal auf:

LOAD DUMP

Sie sehen, daß wir auch hier keine (Kennung) Extension angeben müssen bzw. dürfen. LOAD hängt automatisch die Extension ".HEX" an. Auf dem Bildschirm erscheint:

FIRST ADDRESS 0100
LAST ADDRESS 0212
BYTES READ 0113
RECORDS WRITTEN 03

Jetzt ist es endlich soweit: Sie haben eine COM-Datei erstellt, die Datei DUMP.COM, das wir unter CP/M einfach mittels Eingabe von

DUMP

aufrufen können. Aber es ist nicht ganz einfach, COM-Dateien zu programmieren, da man sich dann auch sehr gut im CCP und dem BDOS auskennen muß, weil auf diese Routinen zurückzugreifen ist.

Ich will aber noch einige Informationen über den Assembler loswerden, damit Sie ihn auch richtig bedienen können (wenn Sie es wollen).

Die Quellprogramme für den Assembler müssen folgende Syntax aufweisen:

<Zeilennummer><Marke>:<Kommando><Argument>;<Kommentar>

Die <Zeilennummer> ist nur optional, d.h. muß nicht vorhanden sein (so in unserer Beispieldatei). Sie wird vom Assembler sowieso überlesen und nur vorgesehen, weil einige Editoren diese Zeilennummern automatisch einfügen, wie beispielsweise der Editor ED auf der CP/M-Diskette. Die <Marke>: muß natürlich auch nicht vorkommen, kann aber an dieser Stelle stehen. Unbedingt vorhanden sein muß in einer Zeile, die nicht ausschließlich aus einem Kommentar besteht, der <Befehl> und das <Argument>. Daran schließt sich dann eventuell noch der ;<Kommentar> an.

Der Assembler wandelt vor der Kodierung alle Kleinbuchstaben in Großbuchstaben um, so wie Sie es von CP/M ohnehin schon gewohnt sind. Dies vereinfacht die Programmierung schon ungemein. Ausgeschlossen von dieser Regel sind Kleinbuchsta-

ben, die in Anführungszeichen stehen und einen fixen Text darstellen, wie beispielsweise die Anweisung:

DB 'File Dump Version 1.4$'

Die einzelnen Komponenten der Assemblerzeile müssen wenigstens durch ein Leerzeichen getrennt sein. Es ist üblich, die <Marke> ganz links beginnen zu lassen und dann mittels eines Tabulators in der Textverarbeitung, mit der Sie die Assembler-Quellcodes schreiben, die verschiedenen Komponenten anzuspringen. Dies ist zwar, wie bereits erwähnt, nicht unbedingt nötig, dient aber der Übersichtlichkeit des Programmes ungemein, da man dann Gleiches immer unter Gleichem stehen hat. Ein Beispiel:

```
DISKR:  ;READ DISK FILE RECORD
        PUSH H! PUSH D! PUSH B
        LXI  D,FCB
        MVI  C,READF
        CALL BDOS
        POP B! POP D! POP H
        RET
```

Diese Passage finden Sie in unserem DUMP-Programm. DISKR ist ein Label, das durch einen Doppelpunkt markiert wird. Es folgt dann der <Befehl> und das <Argument>, immer untereinander. In einer Zeile haben wir auch einen Kommentar, geführt durch ein Semikolon ";".

Sie können in dieser kleinen Routine auch eine weitere Besonderheit des ASM-Assemblers erkennen: Verschiedene Assembler-Kommandos lassen sich durch das Ausrufungszeichen trennen.

Man kann bei ASM Kommentare mit einem Semikolon kennzeichnen oder eine ganze Zeile als Kommentar definieren, indem man in der ersten Spalte mit einem Stern beginnt. Dies wurde von den Entwicklern von ASM eingeführt, um eine gewisse Kompatibilität zu den bestehenden 8080-Assemblern zu wahren. So kann man schön einzelne Routinen in den Assembler-Programmen mit einer Kopfzeile versehen, etwa:

```
**********************************************************
***          Read Disk File Record                     ***
**********************************************************
```

wäre ein Beispiel hierfür.

Achten Sie bei der Verwendung von Labels darauf, daß Sie keine festen Schlüsselwörter integrieren. Schlüsselwörter sind beispielsweise die Mnemocodes des 8080-Prozessors, also Wörter wie MVI, MOV, STA etc.

Im ASM-Assembler können Sie, wie in vielen anderen Assemblern auch, die momentane Position des Program Counters PC während des Assemblier-Vorganges bestimmen. Beispielsweise wird dies automatisch bei Programmlabels gemacht, wie in folgendem Beispiel:

posit: EQU $

Argumente, die im ASM vorkommen, können durch verschiedene arithmetische Operanden verknüpft werden, so beispielsweise:

MVI A,12+2*3

Arithmetische Operanden sind:

+X	Positive Zahl
-X	Negative Zahl, entspricht 0-X (Immer 16 Bit !)
X+Y	Addition zweier 16-Bit-Werte
X-Y	Subtraktion zweier 16-Bit-Werte
X*Y	Produkt von X*Y
X/Y	Division der Argumente
X MOD Y	Restfunktion der Division X/Y

Weiterhin können diese zwei Schiebebefehle verwendet werden:

X SHL Y	Verschiebt den 16-Bit-Wert X um Y Positionen nach links. Herausgeschobene Bits gehen verloren.
X SHR Y	Verschiebt den 16-Bit-Wert X um Y Postionen nach rechts. Herausgeschobene Bits gehen verloren.

Ferner gibt es noch die logischen Operatoren:

NOT X	Logische Negation des Argumentes X
X AND Y	Logische Und-Verknüpfung der Argumente X und Y
X OR Y	Logische Oder-Verknüpfung der Argumente X und Y
X XOR Y	Exklusiv-Oder-Verknüpfung der Argumente X und Y

All diese Möglichkeiten erscheinen Ihnen auf den ersten Blick vielleicht als überflüssig. Sind sie aber nicht, ganz im Gegenteil: Sie sind sogar sehr nützlich. Wenn Sie beispielsweise den Akkumulator mit dem höherwertigen Byte einer Adressemarke laden wollen, so können Sie das ganz einfach

tun, indem Sie diese Marke um 8 Bits logisch nach rechts verschieben:

MVI A,Label SHR 8

Wenn Sie lediglich die unteren 8 Bits haben wollen, so müssen Sie ausdrücklich die oberen 8 Bits ausblenden, weil Sie sonst das 8-Bit-Register, den Akkumulator, mit einem 16-Bit-Wert laden würden. Diese Ausblendung sieht so aus:

MVI A,Label AND 0FFH

Diese Anwendung wurde als Beispiel genannt, weil sie wohl am häufigsten vorkommen dürfte, es sind natürlich tausende von anderen Anwendungen denkbar.

Weiterhin existieren noch die Pseudo-Opcodes

ORG, EQU, END, DS, DB, DW, SET, IF und ENDIF.

(Pseudo-Opcodes sind Assembler-Kommandos, die während des Assemblier-Vorganges vom Assembler gelesen und verarbeitet werden. Der Name "Pseudo-Opcodes" deshalb, weil diese Befehle, wie die normalen Opcodes, im Assembler-Programm vorkommen, nicht aber assembliert und von der CPU verstanden werden können.)

Wir wollen die einzelnen Pseudo-Opcodes nicht näher erläutern, lediglich kurz die Wirkung und Funktionsweise erwähnen, da dies nicht die Aufgabe eines CP/M-Buches sein sollte.

ORG <Startadresse>

Dieses Kommando definiert die Startadresse des zu assemblierenden Programmes. ORG sollte immer das erste Kommando in einem Programm sein.

EQU <Wert>

EQU ordnet einem Label (einer Marke) einen festen Wert zu. Der <Wert> kann auch ein arithmetischer Ausdruck sein.

Beispiel: **Diskan: EQU Diskr+055H**

DS

Dieses Kommando hält im Programm einen definierten Bereich frei, um beispielsweise Daten dort abzulegen. Der Programm-Counter wird entsprechend erhöht.

Beispiel: **Diskm: DS 100**

In diesem Beispiel werden 100 Bytes frei gehalten, Startadresse ist Diskm, Endadresse ist Diskm+99. Das nächste Kommando beginnt bei Diskm+100.

DB

Hier werden einzelne zu definierende Bytes im Speicher abgelegt. Die einzelnen Bytes müssen durch Komma getrennt sein. Es können auch Zeichenketten vorkommen.

Beispiele: **Disktxt: DB "Bitte Diskette einlegen"**
Diskt2: DB 23,32,0,3,122

DW

DW definiert Wörter, also 16-Bit-Werte. Auch hier können die einzelnen 16-Bit-Werte durch Komma voneinander getrennt werden.

Beispiel: **Diskadr: DW 03AB9H,markel**

END

END definiert das Ende des zu assemblierenden Programmes. Für Sonderzwecke ist es möglich, hier auch eine Startadresse für die Abarbeitung des assemblierten Programmes anzugeben.

SET

Mit dem Kommando SET wird, ähnlich dem EQU-Kommando, ebenfalls einer Marke ein Wert zugeordnet. Allerdings mit einem Unterschied: Während beim EQU-Kommando die Zuordnung fix ist, können durch das SET-Kommando definierte Marken während des Assemblierens noch verändert werden. Durch SET definierte Marken werden während des Assemblierens auch nicht in der linken Spalte angezeigt.

Beispiel: **Markel: SET Alfa**
Markel: SET Markel + 32

Doch damit sind wir noch nicht an die Leistungsgrenzen von ASM gestoßen. So wie viele andere leistungsstarke Assembler verfügt auch ASM über die Möglichkeit, konditioniert zu assemblieren, d.h. eine bestimmte Assemblerpassage nur unter bestimmten Bedingungen zu dekodieren und in Maschinensprache umzuwandeln. Dies erreichen Sie, indem Sie sich der Kommandos IF und ENDIF bedienen. (Allerdings werden diese Kommandos erfahrungsgemäß nur selten genutzt.)

Nehmen wir einmal an, Sie wollen ein Programm schreiben, um zwei Werte einzulesen, diese dann zu addieren bzw. miteinander zu multiplizieren. Der Algorithmus ist für beide Programme gleich. Sie können nun ein Programm schreiben, das während des Assemblierens ein Flag abfragt, ob das Programm addieren oder multiplizieren soll.

```
;
Multi  EQU  0 ;Programm soll addieren
ORG 0100H
;
;Lies Werte ein
CALL Einlesen
;
IF Multi
  CALL Multir ;Multiplikationsroutine aufrufen
ENDIF
IF NOT Multi
  CALL Addir  ;Additionsroutine aufrufen
ENDIF
CALL Ausgabe
END
```

Wollen Sie die Routine nun multiplizieren lassen, so setzen Sie in der ersten Zeile den Wert für Multi lediglich auf NOT 0, das ist alles.

Nachdem wir nun den Assembler so ausführlich behandelt haben, wollen wir uns noch einmal kurz dem Aufruf des ASM zuwenden:

Wie Sie sich erinnern, müssen Sie beim Aufruf von ASM nicht die Extension ".ASM" mit angeben. Es werden die beiden Dateien <Dateiname>.PRN und <Dateiname>.HEX auf Diskette erzeugt. Dies kann man aber unterdrücken! Ferner kann man angeben, von welchem Laufwerk die zu assemblierende Datei geladen werden soll, bzw. auf welchem File die zwei zu erzeugenden Files abgelegt werden sollen. Die Syntax des ASM-Kommandos sieht wie folgt aus:

ASM <Datei>.<ASM-Datei><HEX-Datei><PRN-Datei>

Außer dem Dateinamen <Datei> sind alle Angaben optional, müssen aber in oben angegebener Reihenfolge erfolgen, falls sie gemacht werden. Folgende Optionen sind nun möglich:

Für ASM-Datei: A oder B.
A oder B ist die Laufwerksangabe der Quelldatei.

Für HEX-File: A, B oder Z.
Auch hier stehen A und B wieder als Laufwerksangabe, auf dem die Datei unter dem Namen <Dateiname>.HEX abgelegt werden soll. Ferner besteht die Möglichkeit, den Assembler anzuweisen, gar keine HEX-Datei zu erzeugen. Dies ist zum Beispiel dann sinnvoll, wenn Sie nur einen Probelauf machen wollen,

um beispielsweise die Syntax überprüfen zu lassen. Dann geben Sie an der zweiten Stelle als Option ein 'Z'.

Für PRN-Datei: A, B, X oder Z.
A und B ist wieder Laufwerksangabe, auf dem das Datei <Dateiname>.PRN erzeugt werden soll. Dieses Protokoll kann aber auch durch Angabe von 'Z' unterdrückt werden (s.o.). Die Protokolldatei können Sie sich außerdem auch auf dem Bildschirm mir der Option 'X' ausgeben lassen.

Nehmen wir einmal an, Sie wollen die Quelldatei mit Namen Asterix von Laufwerk B laden, die Protokolldatei auf dem Bildschirm und die Hex-Datei überhaupt nicht erhalten, beispielsweise um den Assembler die Syntax Ihres Programmes überprüfen zu lassen, dann geben Sie folgendes ein:

ASM Asterix.BZX

Sie sehen, es ist sehr einfach. Doch lassen Sie uns nun das Kapitel Assembler abschließen, sicherlich haben Sie nun genug Informationen über den ASM-Assembler erhalten.

VII.7 Arbeiten mit SUBMIT und XSUB

Häufig kommt es vor, daß man bestimmte Befehlsfolgen immer und immer wieder eingeben muß. Stellen Sie sich vor, Sie wollen eine bestimmte Quelldatei assemblieren, dann die HEX-Datei in eine COM-Datei umwandeln, und schließlich genau diese neue COM-Datei starten, um es auszuprobieren.

Genau dann ist es empfehlenswert, eine SUBMIT-Datei zu erstellen, die alle nötigen Befehle enthält und hintereinander automatisch ausführt. SUBMIT heißt nichts anderes als übergeben: Das Programm SUBMIT übergibt CCP, dem Console Command Processor, der für die Eingabe über Tastatur und die Ausführung der eingegebenen Kommandos zuständig ist, diese Kette von Kommandos. Dies geschieht so:

Wenn Sie SUBMIT sagen, welche Datei als SUBMIT-Datei übergeben werden soll, so erstellt SUBMIT eine Zwischendatei $$$.SUB. In dieser Zwischendatei stehen all die Dinge der SUBMIT-Datei, mit einigen Ergänzungen. Bevor CCP einen Befehl von der Tastatur übernimmt, wird nachgesehen, ob es nicht die temporäre Datei $$$.SUB auf dem angemeldeten Laufwerk gibt. Sollte dies der Fall sein, so wird eine Befehlszeile in den Buffer von CCP geladen, und diese Zeile aus der Datei gelöscht. Nachdem die Befehlszeile kopiert worden ist, wird der Befehl ausgeführt. Dies wird so lange wiederholt, bis die Datei $$$.SUB leer geworden ist, erst dann wird wieder ein Kommando von der Tastatur eingelesen.

(Da Sie in der zumeist englischen Fachpresse und -literatur niemals den Begriff *Datei* finden werden, wollen wir nun diesen Begriff durch *File* ersetzen).

Wie Sie wissen, wird Ihnen beim DIR-Kommando leider nicht ausgegeben, wieviel Platz noch auf der Diskette ist. Diese Information erhalten Sie über das STAT-Kommando. Wir wollen

nun eine SUBMIT-Datei erstellen, die das DIR-Kommando und das STAT-Kommando zusammenfaßt, um uns mit einem Befehl Auskunft über Kapazität und Inhalt der Diskette zu verschaffen. Dazu erstellen wir mit einer Textverarbeitung, beispielsweise mittels ED, diese SUBMIT-Datei und nennen Sie DISK.SUB. Alle Submit-Dateien müssen das Kürzel ".SUB" tragen, dies ist oberstes Gebot.

Also gut, unsere erste Submit-Datei sieht also so aus:

STAT
DIR

Zugegeben: Es ist eine sehr kurze SUBMIT-Datei, aber es ist noch kein Meister vom Himmel gefallen! Lassen Sie uns diese erste SUBMIT-Datei einmal starten mit:

SUBMIT DISK

Sie sehen, daß wir nicht die Extension ".SUB" angeben dürfen. Unser Laufwerk hört sich nun sehr vielbeschäftigt an, aber wir wissen ja auch, was alles gemacht wird. (Erstellen der $$$.SUB-Datei, kopieren des Inhaltes von DISK.SUB, zeilenweises Löschen der Kommandos aus der Datei $$$.SUB etc.) Und es klappt.

Doch erinnern wir uns, was wir eigentlich geplant hatten. Wir wollten ein File assemblieren, es mittels LOAD-Kommando zum COM-File machen und anschließend starten. Diese Reihenfolge der Kommandos sähe etwa so aus:

ASM <Filename>.<Options>
LOAD <Filename>
<Filename>

Die Begriffe <Filename> und <Options> stehen hier praktisch als Platzhalter, als Variablen - doch wie kann man diese Variablen füllen? Die Entwickler des SUBMIT-Programmes haben die Problematik erkannt und sich etwas einfallen lassen, um die SUBMIT-Dateien etwas komfortabler zu gestalten. Sie können beim Aufruf der SUBMIT-Datei diese Vorgaben übergeben, indem Sie diese Vorgaben durch mindestens ein Leerzeichen voneinander trennen. Diese Vorgaben können einfache Zahlen sein oder aber auch komplexe Begriffe wie etwa Filenamen. Die Vorgaben werden durchnumeriert, beginnend bei eins. In der SUBMIT-Datei muß man diese Vorgaben mit einem Dollarzeichen ($) markieren, direkt gefolgt von der Nummer der Vorgabe. Dabei können einzelne Vorgaben beliebig oft in der SUBMIT-Datei wiederholt werden. Bei der Erstellung des Files $$$.SUB werden dann die Platzhalter automatisch durch die Vorgaben ersetzt, so daß CCP diese selbstverständlich abarbeiten kann. Unsere SUBMIT-Datei sieht nun so aus:

ASM $1.$2
LOAD $1
$1 $3

Sie sehen, daß die erste Eingabe, also die erste Vorgabe, in der SUBMIT-Datei dreimal vorkommt. Wenn wir diese SUBMIT-Datei beispielsweise mit unserem Dump-Programm starten und kein Protokoll-File wollen (ein HEX-File muß erstellt werden, weil LOAD dieses ja braucht), so könnte der Aufruf so aussehen, wenn wir die SUBMIT-Datei unter dem Namen "ASSEM.SUB" gespeichert haben:

SUBMIT ASSEM DUMP AAZ DUMP.COM

Die letzte Angabe weist darauf hin, welches File ausgedumpt werden soll, in unserem Beispiel DUMP.COM selbst. Wenn Sie es ausprobiert haben und alles klappt, dann haben Sie den ersten Schritt zur Benutzung der SUBMIT-Dateien getan. Erster Schritt deswegen, weil SUBMIT noch mehr kann!

Vielleicht haben Sie sich schon gefragt: Was tun, wenn ein Dollarzeichen in der SUBMIT-Datei vorkommen soll oder muß? Ganz einfach: Es wird das gemacht, was man in einem solchen Fall meistens in der Computerei macht, man schreibt zwei Dollarzeichen hintereinander. Nehmen wir einmal an, Sie wollten alle temporären Files löschen, so führen Sie dies aus über Console mit:

ERA *.$$$

In einer SUBMIT-Datei sieht das etwas erschreckender aus:

ERA *.$$$$$$

- bedeutet aber dasselbe. Da Dollarzeichen aber in der Regel recht selten vorkommen, ist der zusätzliche Arbeitsaufwand nur minimal.

Wenn eine Befehlszeile aus dem File $$$.SUB in den Speicher von CCP kopiert wird, wird diese Zeile auch erst einmal auf dem Bildschirm dargestellt. Danach wird die Tastatur

abgefragt, ob eine Taste betätigt worden ist. Ist dies der Fall, oder wird das Kommando im Befehlsspeicher nicht oder nicht richtig ausgeführt, so bricht SUBMIT den Befehlsablauf ab und löscht das FILE $$$.SUB.

Wenn eine Zeile in der SUBMIT-Datei mit einem Zeichen beginnt, das normalerweise nicht in einem Filenamen vorkommen darf, so wird diese Zeile zwar auf dem Bildschirm angezeigt, nicht aber ausgeführt. Auf diese Art und Weise können Sie leicht Kommentare oder Anweisungen für den Benutzer auf den Bildschirm zaubern. Beispielsweise wäre es möglich, in unserer SUBMIT-Datei ASSEM.SUB folgendes zu ergänzen:

```
:Es wird jetzt das File assembliert.
:Bitte gedulden Sie sich ein wenig !!
ASM $1.$2
LOAD $1
$1 $3
```

Und schon ist diese SUBMIT-Datei bei der Ausführung ein wenig aufgelockert. Zeichen, die dieses bewirken sind:



Bei SUBMIT beschränken sich die Vorgaben allerdings auf's CCP, d.h. man kann keine Vorgaben für ein aufgerufenes Programm wie etwa PIP oder ED machen. Manchmal wäre aber auch dies wünschenswert, beispielsweise um bestimmte Dateien mittels PIP zu kopieren oder bestimmte Anweisungen an ED zu übergeben. Auch dieses "Manko" ist zu beheben. Hierzu existiert das Programm XSUB, das alle Eingaben über die Konsole abfängt und die Anfragen aus der $$$.SUB-Datei übernimmt. Wenn wir beispielsweise bei PIP Angaben machen

wollen (das geht natürlich auch direkt), so könnte dies so aussehen:

XSUB
PIP
b:=a:*.COM

In der letzten Zeile müssen Sie einmal RETURN betätigen, um die transiente Routine PIP zu verlassen. In PIP ist diese Anwendung noch nicht so sinnvoll, weil man das Kommando direkt hätte anders gestalten können, etwa: **PIP b:=a:*.COM** um alle COM-Files von Laufwerk A: auf Laufwerk B: zu kopieren, beim ED allerdings wird die Anwendung sehr interessant. Sie sehen, daß es reicht, in der ersten Zeile XSUB einmal aufzurufen. Angaben, die sonst über Tastatur gemacht werden, kommen nun aus dem File $$$.SUB.

In den alten CP/M-2.2-Versionen war noch ein Fehler im SUBMIT-Programm versteckt. Wenn man Daten mittels XSUB übertrug, so konnte man keine Steuerzeichen übergeben, obwohl dies im Handbuch so vorgesehen war. Dies ist sehr ärgerlich, da man beispielsweise bei PIP oder ED nicht selten das Steuerzeichen ^Z senden will oder muß. Da aber dieser Fehler in Ihrer CP/M 2.2-Version behoben ist, müssen Sie sich auch keine weiteren Gedanken darüber machen.

Ich hoffe, Sie haben die Anwendung und Handhabung von SUBMIT-Dateien soweit verstanden, daß Sie demnächst bei entsprechenden Bedarf eine eigene SUBMIT-Datei erstellen können.

Wir haben jetzt den Assembler und die SUBMIT-Dateien besprochen. Kurz erwähnt werden sollten noch ein Disassembler, der sich auch auf der Diskette befindet. Sie

rufen diesen Disassembler mit **DDT** auf. Wollen Sie beispielsweise das Kommando-File FILECOPY.COM einmal disassembliert ausgedruckt bekommen, so geben Sie einfach ein:

DDT filecopy.com

Bevor Sie <RETURN> drücken, geben Sie natürlich noch mit <Ctrl>/P das Kommando, daß alles auf Drucker protokolliert werden soll.

Disassemblieren bedeutet das Gegenteil von Assemblieren: Es werden die Maschinencodes in menschenwürdigere Mnemocodes umgewandelt.

VII.8 Die Speicherverteilung

Wir haben bereits einige Male Schlagwörter wie BDOS, CCP, BIOS etc. erwähnt. Es dürfte den meisten von Ihnen auch nicht verborgen geblieben sein, daß diese Begriffe sehr wichtig sind, zumal sie doch auf dem Buchumschlag so vielverprechend abgedruckt sind. Sicherlich haben sich schon einige Leser Gedanken darüber gemacht, wo sich dies alles im Speicher des Rechners befindet. Lassen Sie uns eben repetieren, was wir unter den einzelnen Begriffen zu verstehen haben:

CCP

CCP steht für Console Command Processor. Dieser Bereich regelt die Eingabe von Tastatur und beinhaltet die residenten Befehle, also: DIR, ERA, REN, SAVE, TYPE und USER.

BDOS

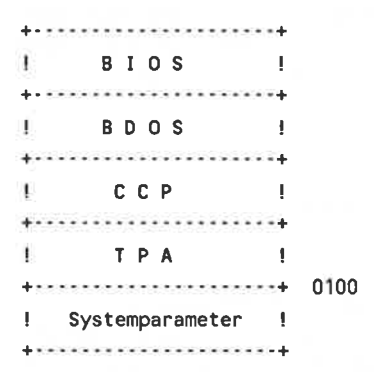
BDOS steht für Basic Disk Output System. Dieser Teil regelt den Datenaustausch zwischen Rechner und Diskettenlaufwerken.

BIOS

Basic Input/Output System soll BIOS abkürzen. BIOS und BDOS arbeiten eng zusammen, warum man die beiden Partner auch häufig unter einem zusammenfassenden Namen findet: FDOS.

TPA

Man nennt den Anwenderprogrammbereich im Speicher des Rechners TPA. Die auszuführenden Programme und die Daten, die mit diesen Programmen bearbeitet werden sollen, werden in diesem Speicher verwaltet.



So sieht im groben die Speichereinteilung von CP/M aus. Der Benutzerbereich (TPA) beginnt bei Adresse $0100.

Da die Erklärung, wie man unter CP/M programmiert, den in diesem Buch vorgesehenen Rahmen sprengen würde, wollen wir nur kurz auf die Programmierung unter CP/M unter Zuhilfenahme der BDOS- und BIOS-Routinen eingehen.

Das BDOS beinhaltet eine Sammlung von Unterprogrammen, die das grundlegende Arbeiten mit Dateien auf Diskette und einigen Peripheriegeräten ermöglicht. Da CP/M schon "einige Jahre auf dem Buckel" hat, existieren beispielsweise auch noch Routinen zur Ein- und Ausgabe von und auf Lochkarte. Sicherlich wird davon auch heute noch kräftig Gebrauch

gemacht, auf unserem CPC können wir solche Routinen aber getrost vernachlässigen.

Um eine BDOS-Routine aufzurufen, muß man folgende standardisierten Vereinbarungen befolgen:

Im Register C des Prozessors wird die Kodenummer der gewünschten Unterroutine an BDOS übergeben. Im Register E bzw. bei 16-Bit-Werten im Registerpaar DE werden der Unterroutine, wenn nötig, Daten übermittelt. Wenn man die Register entsprechend versorgt hat, springt man BDOS an Adresse 5 durch einen Unterprogrammbefehl (CALL) an. BDOS ermittelt dann anhand des C-Registers die Adresse, an der die gewünschte Routine steht. Sie sehen, daß Programme auf diese Art und Weise auf wirklich allen Rechnern lauffähig gemacht werden können, da man fixe Regeln beachtet. Alle rechnerabhängigen Dinge wie Ein- und Ausgabe von Tastatur und Bildschirm werden von für den Rechner speziell entwickelten BDOS-Routinen erledigt. So muß man bei der Entwicklung eines neuen Rechners "nur" diese Routinen schreiben, und schon laufen alle CP/M-Programme auf diesem Rechner.

Zwei Voraussetzungen müssen natürlich gegeben sein. Der Rechner muß einen 8080-Prozessor oder einen kompatiblen besitzen, und es muß möglich sein, diese vorhandenen Programme auch einzulesen. Sollte zweiteres nicht möglich sein, so könnten immerhin theoretisch die CP/M-Programme auf diesem Rechner fahren. Allerdings ist es erfahrungsgemäß das kleinere Problem, vorhandene Programme zu transferieren. Dies kann a) durch Programmierung des Controllers geschehen (s.o.), oder b) indem man durch ein relativ einfaches Programm die vorhandenen Programme über eine gemeinsame Schnittstelle zweier Rechner, beispielsweise über den RS-232, übermittelt. Aber selbst wenn keine gemeinsame Schnittstelle existiert, so finden Tüftler immer einen Weg; bei-

spielsweise kann der Joystick-Port als Eingangsport zur Übermittlung eines Programmes durchaus sehr dienlich sein.

Ein Kollege aus dem Hause DATA BECKER hat sogar Daten vom CBM 8296 zum CPC übertragen, indem er die Daten am CBM 8296 über den User-Port über zwei Kabel an einer Brücke auf der CPC-Platine angeschlossen hat, auf der die verschiedenen Hersteller des CPC "markiert" werden können. Je nach durchtrennter Lötstelle (markierter Brücke) wird dann ein anderer Hersteller beim Kaltstart angezeigt.

Doch zurück zu BDOS. Sollte BDOS Werte aus der BDOS-Routine an das Hauptprogramm zurückliefern, so werden sich 8-Bit-Ergebnisse im Akkumulator des Prozessors auffinden, 16-Bit-Werte stehen als Rückgabe im HL-Registerpaar. Bei 16-Bit-Rückgaben findet sich zusätzlich das niederwertige Bit ebenfalls im Akku wieder, das höherwertige Bit befindet sich im B-Register.

Wollen Sie beispielsweise ein Zeichen auf die Konsole ausgeben, so benutzt man dazu die BDOS-Routine Nr. 2. Wir geben nun (in Z-80-Mnemonik) ein Fragezeichen aus:

```
LD   C,2        ;Konsolenausgabe
LD   E,"?"      ;Fragezeichen laden
CALL 5          ;BDOS anspringen
```

Wenn Sie wirklich ernsthaft in Maschinensprache unter CP/M programmieren wollen, so existiert speziell für Programmierer unter CP/M Fachliteratur. Wir geben Ihnen hier eine Liste der BDOS-Routinen mit Ein- und Ausgabeparameter zum Experimentieren. Zum seriösen Programmieren benötigt man sicherlich einiges mehr an Wissen über die einzelnen Routinen.

Routinenname	C	übernimmt	liefert
Warmstart auslösen	0		
Konsoleneingabe	1		A:Eingabe
Konsolenausgabe	2	E:Zeichen	
Lochstreifen lesen	3		A:Zeichen
Lochstreifen stanzen	4	E:Zeichen	
Zeichen an Drucker	5	E:Zeichen	
Direkte Konsolen Ein-	6	E: 255	A:0
und Ausgabe		E:Zeichen	A:Zeichen
IOBYTE abfragen	7		A:IOBYTE
IOBYTE setzen	8	E:IOBYTE	
Eine Zeichenkette ausg.	9	DE:=>Zeichenkette	
Eingabe aus Buffer	10	DE:=>Buffer	A:Zeichen
Routinenname	C	übernimmt	liefert
Konsolenstatus	11		A:=0 (keine
			Taste)
CP/M Version	12		HL:Version
Diskettensystemreset	13		
Bezugslaufwerk festl.	14	E:LW-Nummer	
Datei eröffnen	15	DE:Infos	A:255 (Error)
Datei schließen	16	DE:Infos	A:255 (Error)
Ersten Eintrag suchen	17	DE:Infos	A:255 (Error)
		sonst:	A:Zeichen
Folgeeintrag	18		A:255/Zeichen
Datei löschen	19	DE:Infos	A:255 (Error)
Aufzeichnung lesen	20	DE:Infos	A:0 (OK)
Aufzeichnung schreiben	21	DE:Infos	A:0 (OK)
Datei erzeugen	22	DE:Infos	A:255 (Error)
Datei umbenennen	23	DE:Infos	A:255 (Error)
aktive Laufwerke ermit.	24		HL:LW-Vektor
Bezugslaufwerk ermit.	25		A:Laufw.Nr.
Datenbuffer fixieren	26	DE:Buffer	
Belegungstabelle ermit.	27		HL:Adresse

Bezugslaufwerk schützen	28		
Geschütze Laufwerke	29		HL:LW-Vektor
Dateioptionen setzen	30	DE:Infos	A:255 (Error)
Diskparameter ermit.	31		HL:Adresse
Benutzernr. verwalten	32	E:255	A:Nummer
		E:Nummer	
Aufzeichnung lesen	33	DE:Infos	A:0 (OK)
Aufzeichnung schreiben	34	DE:Infos	A:0 (OK)
Dateigröße ermitteln	35	DE:Infos	(im Buffer)
Beschreiber setzen	36	DE:Infos	(im Buffer)
Laufwerk rücksetzen	37	DE:LW-Vektor	A:0

Diese Tabelle soll nur anzeigen, wie in etwa eine Programmierung mit BDOS aussehen kann, und welche Möglichkeiten Sie mit BDOS haben. Wie bereits erwähnt, reicht sie selbstverständlich nicht zur Programmierung aus, da interne Kenntnisse der einzelnen Routinen nötig sind.

Beinhaltet BDOS Routinen wie "Bezugslaufwerk schützen", so sind die Aufgaben von BIOS noch spezieller. BIOS besteht ebenso wie BDOS aus einer Liste an Routinen - die aber ausschließlich der Ein- und Ausgabe von Daten dienen. Teilweise findet man im BDOS Routinen, die dieselben Aufgaben haben wie Routinen im BIOS, so beispielsweise Konsolenstatus ermitteln.

BIOS trägt in ganz entscheidendem Maße zum Erfolg von CP/M bei, denn es ermöglicht den CP/M-Programmen, die ausgesprochen rechnerabhängigen Ein- und Ausgaben von und auf Konsole mittels spezieller Routinen durchführen zu lassen.

BIOS ist in vier grobe Bereiche untergliedert:

- *Schnittstelle zum BDOS und den CP/M-Programmen (beispielsweise bei den SUBMIT-Dateien),*
- *Schnittstelle zu den externen Speichermedien (Floppys),*
- *Ein- und Ausgabe über die durch BDOS angesprochene Peripherie,*
- *Pufferung von Daten, die dann entsprechend rechtzeitig zur Verfügung gestellt werden (SUBMIT-Dateien).*

Wie Sie wissen, sind BDOS und BIOS im Speicher verschiebbar. Die Routinen im BDOS werden durch Übergabe der Routinennummer im C-Register angesprochen - aufgerufen wird dann die fixe Speicheradresse &0005. Beim Aufruf von BIOS-Routinen sieht das ein wenig anders aus: Es gibt eine Sprungtabelle, deren Reihenfolge zwar vorgeschrieben ist, allerdings kann die Startadresse dieser Sprungtabelle verschoben werden. Doch zunächst wollen wir uns diese Sprungtabelle einmal betrachten:

```
00+Offset: JMP BOOT     ;Kaltstart
03+Offset: JMP WBOOT    ;Warmstart
06+Offset: JMP CONST    ;Frage Konsolenstatus ab
09+Offset: JMP CONIN    ;Konsoleneingabe
0C+Offset: JMP CONOUT   ;Konsolenausgabe
0F+Offset: JMP LIST     ;Ausgabe auf Drucker
12+Offset: JMP PUNCH    ;Lochstreifenstanzer
15+Offset: JMP READER   ;Lochstreifenleser
18+Offset: JMP HOME     ;Kopf auf Spur 0
1B+Offset: JMP SELDSK   ;Laufwerk auswählen
1E+Offset: JMP SETTRK   ;Spur setzen
21+Offset: JMP SETSEC   ;Aufzeichnungsabschnitt auswählen
24+Offset: JMP SETDMA   ;Auswählen des Datenpuffers
27+Offset: JMP READ     ;Aufzeichnungsabschnitt lesen
2A+Offset: JMP WRITE    ;Aufzeichnungsabschnitt schreiben
```

```
2D+Offset: JMP LISTS     ;Frage Druckerstatus ab
30+Offset: JMP SECTRAN   ;Aufzeichnungsnummer übersetzen
```

Der Offset steht hier für den Versatz im Speicher, auf den ich noch genauer eingehen will. Wird eine der angegebenen Funktionen nicht benötigt - dies dürfte beim CPC bei den zwei Routinen der Fall sein, die die Lochstreifen betreffen -, so werden die entsprechenden Speicheradressen einfach mit

RET ! NOP ! NOP

aufgefüllt, damit sich der Speicher hinter dieser Routine nicht verschiebt. Ein Anspringen dieser Routine würde also sofort mit einem RETURN quittiert - es passiert also nichts. Um die Startadresse (den Offset) des BIOS zu ermitteln, muß man wissen, daß an Adresse 0 der Grundseite des Systems ein Sprung in den Warmstartvektor (den zweiten Vektor in der Tabelle) des BIOS enthalten ist. Dadurch wird es leicht, die Startadresse zu ermitteln. Wir wollen dies einmal tun. Starten Sie zunächst CP/M, dann laden Sie den Disassembler:

DDT

Um uns den Speicherbereich ab Adresse 0 disassemblieren zu lassen, geben wir folgendes in den Rechner ein:

l0000

Und wir erhalten auch prompt die Antwort von DDT:

```
0000  JMP  AD03
0003  ADD  C
0004  NOP
0005  JMP  8F00
0008  JMP  B982
```

etc. Was uns interessiert, ist also die erste Zeile: Es wird nach Adresse &AD03 gesprungen, die Sprungtabelle von BIOS beginnt also an Adresse &A300. Das ist schon alles, Wissenswerte für unsere Programme. Wollen wir einen Vektor anspringen, so errechnen wir die Einsprungadresse durch:

Adresse:=Sprungnummer * 3 + &A300

Genau wie bei den BDOS-Routinen können Sie auch bei den BIOS-Routinen Parameter übergeben oder entgegennehmen. Sollen Werte an die BIOS-Routinen geliefert werden, so werden 8-Bit-Werte im C-Register übergeben, 16-Bit-Werte werden im Registerpaar BC erwartet. Sie können hier schon einen Unterschied zu den BDOS-Routinen erkennen, die die Parameter im E-Register bzw. im Registerpaar DE erwarten. Parameter, die die BIOS-Routinen an das Hauptporgramm zurückliefern, werden parallel zu den BDOS-Routinen bei 8-Bit-Werten im Akku, bei 16-Bit-Werten im Registerpaar HL übergeben. Tüftler haben sicherlich schon eine sehr schöne Möglichkeit entdeckt: Da die einzelnen BIOS-Routinen über Vektoren im RAM angesprungen werden, kann man diese natürlich verändern, d.h. auf eine eigene Routine umleiten, die gegebenenfalls dann ins Originalprogramm verzweigt.

Wir wollen eben noch auf die weiteren Eigenschaften der Grundseite (Adresse &0000 bis &0100) des Systems eingehen. Durch das Kommando l0 haben wir uns den Bereich &0000 bis

&0016 disassemblieren lassen. Dabei fallen noch einige weitere JMP-Befehle auf: Besonders wichtig ist der Sprungbefehl an Adresse 5 - es müßte jetzt eigentlich bei Ihnen klingeln, denn an Adresse 5 haben wir doch unseren Unterprogrammaufruf gerichtet, wenn wir eine BDOS-Routine aufrufen wollten! Also muß BDOS bei unserem CPC an Adresse &8F00 liegen, da der Sprungbefehl an Adresse &0005 uns dorthin verweist. Dann wollen wir uns diesen Speicherbereich doch einmal etwas näher ansehen:

I8F00

Damit es nicht so einfach ist, dürfen wir noch ein weiteres Mal springen, und zwar an Adresse &95A2. Lassen Sie uns nicht verzagen, und diesen Sprung auch noch nachvollziehen:

L95A2

Um es kurz zu machen: Es wird noch nach Adresse &9F06 gesprungen, dann an Adresse &9F11. Erst dann wird allmählich mit der Dekodierung des C-Registers begonnen, um die anzuspringende Routine zu ermitteln. Es würde sicherlich zu weit führen, diesen Ablauf weiter zu verfolgen, Sie können dies mit Hilfe von DDT und einer Nachtschicht in Ihrer Stube sehr leicht selber tun.

VII.9 Das SETUP-Kommando

Das SETUP-Kommando gibt es nur unter CP/M 2.2; es ist aber auch für die CPC 6128-Besitzer interessant, da man auf dem 6128 ja auch CP/M 2.2 fahren kann.

Vielleicht haben Sie auch an der Hintergrundfarbe, die standardmäßig bei CP/M eingeschaltet wird, wenig Gefallen gefunden? Oder Sie wünschten sich, daß der CPC sich nicht mit der unpersönlichen Zeile

CP/M 2.2 - Amstrad Consumer Electronics plc.

meldet? Dann ist das Kommando SETUP genau das richtige für Sie. SETUP dient dazu, das CPC-CP/M ganz individuell auf den eigenen Bedarf zuzuschneiden. SETUP ist menügesteuert und fragt die einzelnen Punkte im Dialog am Bildschirm ab. Sie können auf diesen Weg realisieren, daß CP/M sich viel netter Ihnen gegenüber verhält und Sie mit einem freundlichen

Hallo Egon, altes Haus - brauchst Du mich mal wieder?

begrüßt. Doch lassen Sie uns einmal das Kommando SETUP aufrufen. Wir gehen dann die einzelnen Unterpunkte etwas genauer durch.

Nachdem Sie SETUP eingegeben haben, erscheint folgendes auf dem Bildschirm:

SETUP V2.0

**** Initial command buffer: empty**

Is this correct (Y/N):_

Die Frage **Is this correct (Y/N)** wird Ihnen nun bei jeder Komponente des SETUP-Kommandos gestellt. Beantworten Sie diese mit Y für Ja, so werden die angezeigten Parameter übernommen, bei Eingabe von N für Nein können Sie noch einmal korrigieren.

Sie sehen, daß unser "Initial command buffer" leer (empty) ist. Wenn Sie in den Initial command buffer etwas hineinschreiben, so wird dieser Buffer beim Einloggen in CP/M ausgeführt. Wollen Sie beispielsweise nach dem Booten von CP/M erst einmal ein Inhaltsverzeichnis der Systemdiskette auf dem Bildschirm haben, um sich einen Überblick über die transienten Kommandos zu verschaffen, so können Sie das über diesen Initial command buffer realisieren. Sie beantworten also die Frage "Is this correct" mit einem satten "Nein", damit Sie Ihren Initial command buffer anpassen können. Nachdem Sie die Taste N betätigt haben, erscheint auf dem Bildschirm:

Enter new initial command buffer:_

Wir wollen unseren Buffer mit dem Kommando DIR füllen. Damit eine Zeile ausgeführt werden kann, muß diese natürlich auch mit einem Wagenrücklauf beendet werden - wie will man aber einen Wagenrücklauf eingeben? Für alle ASCII-Zeichen, die kleiner als 32 sind, müssen die Kontrollsequenzen ^A bis ^Z, sowie ^(,^/,^),^> und ^- verwendet werden. Um beispielsweise den Wagenrücklauf einzugeben, müssen Sie sich des Zeichens ^M bedienen. (Der Wagenrücklauf hat den ASCII-Code 13 und

der Buchstabe M ist der 13. im Alphabet!), Tragen Sie also folgendes in den Initial command buffer ein:

DIR^M

nachdem Sie die <ENTER>-Taste gedrückt haben, erscheint wieder die Frage **Is this correct (Y/N):_**, worauf Sie jetzt mit einem Y für Ja antworten können.

Als nächstes wird das SIGN-ON-STRING angezeigt und abgefragt. Im SIGN-ON-STRING finden sich eine Menge Steuerzeichen wieder, die das Setzen des 80-Zeichen-Modus und der Hintergrund- und Rahmenfarbe bewirken. Auch hier müssen ASCII-Zeichen kleiner als 32 durch ^<Code> eingegeben werden. Die entsprechenden ASCII-Zeichen für: Setzen der Hintergrundfarbe, Setzen des 80-Zeichen-Modus etc. finden Sie im Bedienerhandbuch in der ASCII-Zeichen-Tabelle. Hier können Sie sich dann Ihre individuelle Promptmeldung kreieren, beispielsweise unser "Hallo Egon...".

Nachdem Sie den Initial command buffer sowie den SIGN-ON-STRING definiert haben, können Sie die Tastatur um- oder zusätzlich mit Zeichen belegen. Hierzu dient der Punkt

Keyboard translations

Die Angaben, die Sie in diesem Menüpunkt machen, sind exakt gleich dem BASIC-Kommando **KEY DEF**. Die entsprechenden Code-Tabellen sowie die Tastaturcodes können Sie dem Handbuch entnehmen. Auch der nächste Menüpunkt, **Keyboard expansions**, hat dieselben Auswirkungen wie das BASIC-Kommando KEY.

Darauf folgend werden Ihnen die Standardbelegungen für die Input/Output-Geräte angezeigt, die Sie ebenfalls editieren können. Es folgen einige technische Angaben zur Floppy (Nachlaufzeit des Motors etc.), sowie Angaben zu den Ein/Ausgabe-Kanälen des Prozessors. Nachdem Sie all diese Angaben über Tastatur quittiert haben, erhalten Sie die Frage:

Do you want to update your System disc (Y/N):_

Beantworten Sie diese Frage mit Y, so werden alle Ihre Angaben auf Diskette gesichert (bei Betätigen der "N"-Taste gehen die Angaben verloren). Die letzte Frage, die von CP/M an Sie gerichtet wird, ist:

Do you want to restart CP/M (Y/N):_

Sie können duch diese Möglichkeit direkt die Auswirkungen Ihrer Änderungen auf dem Bildschirm überprüfen, indem Sie mit Y für "Ja" antworten. Es wird dann CP/M neu gebootet, und sollten Sie beispielsweise den SIGN-ON-STRING oder den Initial command buffer geändert haben, so wird dies sofort ausgeführt.

Sicherlich haben Sie jetzt an CP/M einiges Interesse gefunden. Es ist auch sehr schwer, CP/M ganz auszureizen, besonders dann, wenn man selbst unter CP/M programmieren will. Wir hoffen, daß Ihnen dieses Buch als Hilfsmittel sehr dienlich war und sein wird.

VIII. Korrektur von PIP

VIII.1 Fehlerbeschreibung

Im PIP-Kommando gibt es noch eine Kleinigkeit, die einem zur Weißglut treiben kann. Arbeitet man mit PIP, um Dateien zu kopieren, und wechselt dann die Diskette ohne einen Warmstart mit CTRL-C zu machen, so erhält man die Meldung "BDOS ERROR R/O". Dies ist sehr ärgerlich, da man nun die CTRL-C-Taste betätigen und alles neu eingeben muß.

Da wir uns sehr ungern ärgern und auch wollen, daß Sie sich in Zukunft nicht mehr ärgern müssen, wollen wir Ihnen einen Lösungsvorschlag unterbreiten, mit dem man die Meldung "BDOS ERROR R/O" für immer der Vergangenheit angehören lassen kann. Dies ist sicherlich auch für Sie recht interessant, vor allem, weil Sie dann auch direkt Einblick gewinnen, wie man DDT verwendet und wie man CP/M-COM-Dateien einsehen und verändern kann.

VIII.2 Die Fehlerkorrektur

Den oben beschriebenen Mißstand wollen wir nun verschwinden lassen, indem wir das PIP.COM-Programm ein wenig korrigieren. Am besten ist es, Sie nehmen die Korrektur zunächst *nur* an Ihrer Sicherheitskopie vor - Sie können sich vorstellen, was passiert, wenn Sie einen Fehler machen und keine Kopie von CP/M mehr haben!

Wir wollen nun die CP/M-Funktion Disk-Reset am Anfang aufrufen lassen, damit die evtl. neu eingelegte Diskette initialisiert wird. Dazu müssen wir also in der PIP.COM-Datei rumpfuschen, am besten mit dem DDT.

Geben Sie ein:

```
DDT PIP.COM
```

Das Programm startet, wie jede andere COM-Datei auch, an Adresse $0100. Sie können sich diesen Bereich disassemblieren lassen:

```
-l0100
 0100 JMP 04CE
 0103 RET
```

(Ihre Eingaben sind in Fettschrift)

Der Start des PIP-Programmes liegt also bei 04CE. Wir aber wollen noch ein DISK-Reset dazwischenfügen, weshalb wir diese kleine Routine am Ende des PIP-Programmes anhängen und diese dann am Anfang von PIP anspringen. Das Ende des PIP-Programmes liegt bei $1DB1. Wir geben nun folgende kleine Routine ein:

```
-a1db2
1DB2 MVI C,0D
1DB4 CALL 0005
1DB7 JMP 04CE
1DBA (RETURN drücken)
```

Diese Routine ruft also die DISK-Reset-Routine auf. Nun müssen wir noch am Anfang von PIP sagen, daß diese Routine

zunächst angesprungen werden soll. Das machen wir wie folgt:

```
-a0100
0100 JMP 1DB2
0103 (RETURN drücken)
```

VIII.3 Abspeichern des neuen PIP

Jetzt ist das PIP-Programm korrigiert. Sie sollten jetzt PIP einmal ausprobieren (am besten mit allen möglichen Optionen). Sollte es nicht laut Beschreibung funktionieren, so ist irgendetwas schief gegangen. Doch dann haben Sie (Gott sei Dank!) noch eine Sicherheitskopie von CP/M, die Sie zunächst kopieren und es dann noch einmal versuchen können. (Sie sehen, wie sinnvoll es ist, eine Sicherheitskopie im Schrank zu haben. Nicht nur bei Manipulationen kann sich eine Diskette für immer von Ihnen verabschieden. Stellen Sie sich vor, Ihre kleine Tochter oder Ihr kleiner Sohn benutzt Ihre Diskette als Schaufel im Sandkasten!)

Jetzt müssen wir unser Programm noch auf Diskette neu sichern, da es sich ja jetzt lediglich korrigiert im Speicher befindet. Dazu müssen Sie zunächst DDT verlassen, indem Sie die CTRL-C-Taste betätigen.

Dann wird das Programm mittels dem SAVE-Kommando neu abgespeichert. Dies geschieht auf folgendes Kommando hin:

SAVE 29 PIP.COM

Wenn die Diskette nicht schreibgeschützt ist, dann ist jetzt alles getan und PIP ist korrigiert. Der Fehler wird nicht mehr auftreten.

Sie sehen, daß selbst so große und bedeutende Programme wie das Betriebssystem CP/M von großen Firmen wie Digital Research noch Fehler aufweisen können. Es ist einfach unmöglich, alle Fehler in einem Programm zu eliminieren - und *kein* Programm ist fehlerfrei.

An dieser Stelle wollten wir noch einen weiteren Fehler in den gängigen CP/M 2.2-Versionen erwähnen, samt Lösungsvorschlag; doch siehe da: Dieser Fehler wurde bereits in der Schneider-CP/M-Version eliminiert. Wie Sie sehen sind die Firmen bemüht, bekanntgewordene Fehler sofort zu eliminie-

IX. Alle CP/M-Befehle

IX.1 ASM (CP/M 2.2)

ASM.COM

Erstellt aus einer Quelldatei mit 8080- oder Z 80-Befehlen eine Zieldatei im HEX-Format

Eingabeformat

ASM DATEI
ASM DATEI.BBZ

Beschreibung

Mit diesem Programm wird aus einer Datei mit 8080- oder Z80-Befehlen eine Hexdatei erstellt. Diese sogenannte Objektdatei kann dann mit dem Programm LOAD als transienter Befehl in das System geladen werden.

Anwendung

Das Programm ASM sucht sich immer eine Datei mit der Kennung ASM, so daß die Kennung nicht eingegeben werden muß. Insgesamt können drei Parameter mit eingegeben werden, die da heißen:shp. S steht für Source, womit die Bezeichnung des Laufwerks gemeint ist (A..P) auf dem die Quelldatei steht. H steht für HEX und ist der Platzhalter für die Bezeichnung des Laufwerks, auf dem die Hexdatei erstellt werden soll. Wollen Sie die Erstellung der Hexdatei unterdrücken, geben Sie ein Z ein. P steht für PRN und Sie können an dieser Stelle das Laufwerk angeben, auf dem die PRN-Datei erstellt werden soll. Um diese zu unterdrücken, geben Sie wieder ein Z ein oder, wenn Sie die PRN-Datei auf dem Bildschirm sehen wollen ein X.

IX.2 COPYSYS

COPYSYS.COM

Kopiert die Systemspuren und CP/M3.SYS auf eine Diskette

Eingabeformat:

COPYSYS (CP/M 3.0)

Beschreibung:

Um von einer Diskette CP/M 3.0 zu starten, müssen beide CP/M-Teile vorhanden sein: Der erste Teil auf den Systemspuren, der zweite als Datei CPM3.SYS. Disketten, die nicht zum Booten verwendet werden, brauchen beides nicht.

Anwendung:

Geben Sie COPYSYS und dann ENTER ein. Sie werden gefragt, von welchem Laufwerk kopiert werden soll:

Source drive name (or return for default)

Geben Sie den Buchstaben des Laufwerks ein, in dem Ihre Diskette mit dem CP/M-System liegt (Laufwerk A: oder 1, normalerweise). Ist die Diskette schon im Laufwerk, geben Sie nur ENTER Dann geben Sie die Bezeichnung des Ziellaufwerks ein und drücken wieder ENTER. Haben Sie die Diskette noch nicht eingelegt, werden Sie dazu aufgefordert. Sind die Systemspuren beschrieben, können Sie noch CPM3.SYS kopieren. Sie werden gefragt:

Do you wish to copy CPM3.SYS?

Antworten Sie mit "Y" und die CP/M-Datei wird kopiert.

Anmerkung:

COPYSYS ist das Standard-Kommando zum Kopieren der Systemspuren unter CP/M 3.0. Leider befindet sich diese COM-Datei auf keiner der mitgelieferten CPM-Disketten. Dafür bietet aber das Programmpaket DISCKIT3 die Möglichkeit, Disketten in jedem beliebigen Format zu formatieren, und dann das System zu kopieren.

DISCKIT3 ist ein sehr bedienerfreundliches Programmpaket, das die Firma Schneider speziell für den CPC 6128 entwickelt hat. Leider lagen bei der Erstellung dieses Buches noch keine Unterlagen über dieses Programmpaket vor. Sicher ist nur, daß Sie mit DISCKIT3 problemlos Disketten formatieren und das System CP/M 3.0 kopieren können. Weitere Informationen zu DISCKIT3 entnehmen Sie am besten dem Bedienerhandbuch.

IX.3 DATE (CP/M 3.0)

DATE.COM

Zeigt Datum und Zeit und ermöglicht deren Eingabe

Eingabeformat:

DATE
DATE CONTINUOUS
DATE SET
DATE MM/DD/YY HH:MM:SS

Beschreibung:

Mit diesem Programm können Sie ein Datum und eine Uhrzeit in Ihr System eingeben oder beide Informationen abrufen.

Anwendung:

Geben Sie nur den Befehl DATE ein, bekommen Sie das Datum und die Uhrzeit angezeigt

Tue 05/06/85 23:49:17

Dies ist eine stehende Anzeige. Möchten Sie die Zeit mit einer anderen Uhr vergleichen, geben Sie den Befehl DATE mit der Option "C" ein. Dann sehen Sie die laufende Uhr, können aber in dieser Zeit kein anderes Programm laufen lassen oder irgendwelche Befehle eingeben. Drücken Sie eine beliebige Taste, um das Programm abzubrechen.

Sie können Datum und Zeit auf zwei Wegen eingeben: einmal mit der Option SET, dann erfragt das Programm die Werte hintereinander und Sie können eine Eingabe mit ENTER überspringen. Die andere Möglichkeit sieht so aus:

DATE 05/06/85 23:49:17

Geben Sie eine Zeit ein, die etwa ein bis zwei Minuten hinter der aktuellen liegt. Wenn der von Ihnen eingegebene Zeitpunkt kommt, drücken Sie nur noch die Leertaste, und die Uhr läuft.

Besitzt Ihr Computer keine batteriegepufferte Uhr, und das ist beim Schneider CPC 6128 der Fall, so müssen Sie die Eingabe jedesmal nach dem Starten eingeben. Sie können sich das erleichtern, indem Sie den Befehl DATE SET in die Datei PROFILE.SUB schreiben.

IX.4 DDT

DDT.COM

Debugger-Programm zum Laden, Ausführen, Ändern und Testen von Programmen

Eingabeformat

DDT

DDT Datei.COM

Beschreibung

Das Programm DDT ersetzt den CCP und lädt dann die angegebene Datei in den Arbeitsspeicher. Mit DDT kann die nächste Adresse nach der Endadresse eines Programms (NEXT) und der Programmzähler-Inhalt (PC) abgefragt werden.

Anwendung

Das Programm muß als DDT.COM auf der Diskette verfügbar sein. Wird nur DDT eingegeben, wartet das Programm noch auf die Eingabe eines Dateinamens. Mit der Eingabe Dn wird ein Speicherbereich, beginnend mit der Adresse n auf den Bildschirm ausgegeben. Zum Beenden des Programms benutzt man am Besten den befehl GO.

IX.5 DEVICE (CP/M 3.0)

DEVICE.COM

Zeigt und verändert die Wege zu den Peripherie-Geräten

Eingabeformat:

DEVICE NAMES
DEVICE VALUES
DEVICE LST:=XXXXXX[NOXON,1200]
DEVICE CONOUT:=XXX,XXX
DEVICE CON:[PAGES]
DEVICE CON:[COLUMNS=nn, LINES=mm]

Beschreibung:

Dieses Programm wird benutzt, um die Ein- und Ausgabekanäle des Betriebssystems anzuzeigen und zu verändern. Die drei logischen Kanäle heißen: CON:, LST: und AUX:. Man kann auch einen logischen Kanal auf mehrere Peripheriegeräte lenken. Zum Beispiel können die Daten aus dem Computer sowohl auf den Drucker, wie auch gleichzeitig auf den Bildschirm geschickt werden. Mit DEVICE können Sie auch die Zahl der Zeilen und Spalten auf dem Bildschirm bestimmen.

Anwendung:

Die Namen der Ausgabeeinheiten werden von den Herstellern festgelegt. Geben Sie DEVICE mit der Option NAMES ein, bekommen Sie eine Liste mit den Namen und die Übertragungsraten der einzelnen Geräte. Geben Sie DEVICE VALUES ein, bekommen Sie etwa so eine Anzeige:

Current Assignments:

CONIN: = CRT
CONOUT: = CRT
AUXIN: = LPT
AUXOUT: = LPT
LST: = CEN

CONIN: und CONOUT: bedeuten CONsol INput, Tastatur-Eingabe und CONsole OUTput, Tastatur-Ausgabe. Steht die Bezeichnung CON: allein, bezieht sie sich auf beides. LST: bedeutet LIST DEVICE und spricht den Drucker an.

Mit der PAGE-Option können Sie feststellen, wieviele Zeilen und Spalten zur Zeit auf dem Bildschirm angezeigt werden. Geben Sie in der Form [nnn] eine bestimmte Baudrate ein, wird der Datenverkehr mit dem dazugehörenden Gerät mit dieser Geschwindigkeit abgewickelt. Auf den Befehl DEVICE hin, bekommen Sie die Anzeige von DEVICE NAMES und VALUES zusammen angezeigt.

Beachten Sie, daß das Kommando DEVICE unter CP/M 2.0 durch das Kommando STAT DEV: ersetzt wird (richtiger müßte man sagen, daß man das Kommando STAT unter CP/M 2.2 gesplittet hat, da es zu komplex ist.)

IX.6 DIR (CP/M 2.2 und CP/M 3.0)

DIR.COM und eingebauter Befehl (CP/M 3.0)

Zeigt bestimmte Dateinamen und das Inhaltsverzeichnis

Eingabeformat:

Eingebauter Befehl:

DIR
DIR A:
DIR DATEI.XXX
DIR AB*.*

Transienter Befehl:

DIR[OPTION]
DIR DATEI.XXX[OPTION]
DIR AB*.*[OPTION]

Beschreibung:

DIR ist ein sehr hilfreiches Programm, um Dateien auf einer Diskette oder erst recht auf einer Festplatte wiederzufinden. (Noch ist kein Anbieter für Festplatten für einen der Schneider-Rechner bekannt - leider) Die eingebaute Version arbeitet von jedem Laufwerk und aus jedem Benutzerbereich. Die "Jokerzeichen" "*" und "?" können benutzt werden. Wird keine Option mit eingegeben, werden alle Nicht-System-Dateien des entsprechenden Benutzerbereiches angezeigt.

DIR als transienter Befehl kann erheblich mehr: Dateien mit Datum-Stempel anzeigen, in alphabetischer Reihenfolge sortieren usw. Eine Aufzählung der verschiedenen Optionen,

die in eckigen Klammern eingegeben werden müssen, folgt jetzt. Stehen auch System-Dateien im Inhaltsverzeichnis, erscheint die Meldung:

SYSTEM FILE(S) EXIST

OPTIONEN:

```
Option Funktion
ATT Gib benutzerdefinierte Datei-Attribute aus
DATE Zeige Dateien mit Zeit und Datum
DIR Zeige Nicht-System-Dateien
DRIVE=ALL Zeige Dateien aller Laufwerke
DRIVE=A Zeige Dateien auf Laufwerk A:
DRIVE=(A,B) Zeige Dateien in den angegebenen Laufwerken
EXCLUDE Zeige alle Dateien, außer dem eingegebenen
FF Seitenvorschub vor Ausgabe des Verzeichnisses
FULL Zeige Dateien mit vollständiger Beschreibung
LENGTH=n Überschrift nach n Zeilen
MESSAGE Zeige Laufwerke und USER-Bereiche fortlaufend
NOPAGE Zeige Inhaltsverzeichnis ohne Stopp
NOSORT Zeige Dateien unsortiert
RO Zeige Read Only Dateien
RW Zeige Read/Write Dateien
SIZE Größe jeder Datei mit angeben
SYS Zeige nur System-Dateien
USER=ALL Zeige Dateien aller USER-Bereiche
USER=5 Zeige Dateien des USER-Bereiches
USER=(3,5) Zeige Dateien der USER-Bereiche
```

IX.7 DIRSYS (CP/M 3.0)

Eingebauter Befehl

Zeigt die Dateien an, die zu SYSTEM-Dateien erklärt wurden

Eingabeformat:

DIRSYS
DIRSYS B:
DIRSYS DATEI.XXX
DIRSYS AB?*.*

Beschreibung:

Mit DIRSYS werden alle SYSTEM-Dateien angezeigt. Dies kann zwar auch mit DIR und Optionen erreicht werden, aber DIRSYS ist schneller, da es ein eingebauter Befehl ist. Sie können den Befehl zu:

DIRS

abkürzen.

IX.8 DUMP (CP/M 2.2 und CP/M 3.0)

DUMP.COM

Zeigt Nicht-Text-Dateien in hexadezimal und ASCII

Eingabeformat:

DUMP
DUMP DATEI.XXX

Beschreibung:

DUMP ist besonders für Programmierer interessant, die mit Assembler arbeiten. Textdateien werden mit TYPE auf den Bildschirm oder Drucker gebracht, Dateien in der Art von COM, REL oder OVR mit DUMP.

Unter Kapitel VII. finden Sie detaillierte Beschreibungen zu den COM-Dateien ASM, DDT und DUMP.

IX.9 ERA(SE) (CP/M 2.2 und CP/M 3.0)

Eingebauter Befehl und ERASE.COM

Löscht eine, mehrere oder alle Dateien von der Diskette

Eingabeformat:

Eingebauter Befehl:

ERASE
ERASE DATEI.XXX
ERASE AB?*.*

Transienter Befehl:

ERASE DATEI.XXX[CONFIRM]

Beschreibung:

Auf den Befehl ERASE hin, wird die angegebene Datei gelöscht, wenn sie nicht "Read Only" oder schreibgeschützt ist. Löschen können Sie nur innerhalb des USER-Bereiches. Benutzen Sie "*" oder "?", wiederholt der Befehl die Löschanweisung, und fragt nach, ob wirklich gelöscht werden soll.

Anwendung:

Ohne Optionen eingegeben, kann ERASE von jedem Laufwerk aus arbeiten. Bei Eingaben mit Optionen, ist das Programm ERASE.COM auf dem Laufwerk notwendig. ERASE kann zu ERA abgekürzt werden. Die Option "CONFIRM" kann zu "C" abgekürzt werden.

IX.10 FORMAT (CP/M 2.2 und CP/M 3.0)

FORMAT.COM

Formatiert ein Diskette neu

Eingabeformat:

FORMAT

Beschreibung:

Jede Diskette, die Sie in Ihrem Computer verwenden, muß entsprechend formatiert sein. Das Programm zerstört beim Formatieren alle Daten auf einer Diskette. FORMAT ist kein Standard-CP/M-Programm. Deshalb kann es ein, daß es auf Ihrer System-Diskette nicht vorhanden ist. Suchen Sie dann nach dem Programm COPY.COM und sehen Sie nach, ob Sie damit eine Diskette formatieren können.

Anwendung:

FORMAT arbeitet solange, bis Sie es mit Control C (^C) abbrechen. So können Sie mehrere Diskette hintereinander formatieren.

Anmerkung:

Beim Schneider CPC 6128 wird unter CP/M 3.0 kein FORMAT als COM-Datei angeboten. FORMAT ist als Unterprogramm im Dienstleistungspaket DISCKIT3 integriert. Wollen Sie FORMAT aber trotzdem als COM-Datei haben, beispielsweise um es in SUBMIT-Dateien einzubinden, so können Sie sich die COM-Datei von der CP/M 2.2-Diskette kopieren.

IX. 11GENCOM (CP/M 3.0)

GENCOM.COM

Erstellt eine spezielle Version von CP/M 3.0

Eingabeformat:

GENCOM

Beschreibung:

Wird von Programmierern gebraucht, um eine CP/M-Version anzupassen oder zusätzliche Programmteile in CP/M einzuflechten.

IX.12 GET (CP/M 3.0)

GET.COM

Holt Daten aus einer Datei, statt von der Tastatur

Eingabeformat:

GET CONSOLE INPUT FROM FILE DATEI.XXX
GET CONSOLE INPUT FROM CONSOLE
GET CONSOLE INPUT FROM FILE DATEI.XXX[SYSTEM]
GET FILE DATEI.XXX[NOECHO]

Beschreibung:

Mit dem GET-Befehl können Sie Eingaben oder Befehle statt über die Tastatur einzugeben, aus einer beliebigen Datei kommen lassen.

Anwendung:

Die erste Zeile oben befiehlt CP/M, den nächsten Befehl oder das nächste Programm zu verarbeiten, der oder das über die Tastatur eingegeben wird. Sobald eine Eingabe nötig wird, holt sich CP/M diese aus der Datei DATEI.XXX. Ist das Programm zu Ende oder kein weiterer Befehl mehr in der Datei, kehrt CP/M wieder zur Tastatur-Eingabe zurück.

Der Befehl in der zweiten Zeile weist CP/M an, wieder Befehle von der Tastatur entgegenzunehmen. Dieser Befehl kann in der Datei DATEI.XXX stehen oder vom Benutzer eingegeben werden.

Die dritte Form des Befehls leitet die Tastatur-Eingabe sofort zu der benannten Datei um, damit diese die Tastatur-Befehle lesen und verarbeiten kann. Geben Sie als Option (NOECHO) mit an, werden die gerade ausgeführten Befehle nicht auf dem Bildschirm angezeigt.

IX.13 HELP (CP/M 3.0)

HELP.COM und HELP.HLP

Gibt Hilfestellung zu den einzelnen CP/M-Kommandos, mit Beispielen

Eingabeformat:

HELP
HELP Stichwort
HELP Stichwort Unterbegriff
HELP Stichwort[Option]
HELP .Unterbegriff
HELP [Option]

Beschreibung:

Mit dem Programm HELP.COM bekommen Sie Hilfe und Informationen über die CP/M-Befehle und -Programme. Sie können HELP nicht aufrufen, wenn Sie gerade mit einem anderen Programm arbeiten.

Anwendung:

Wenn Sie HELP aufrufen, bekommen Sie eine Reihe von Stichwörtern angeboten. Sie können sich das für Sie interessante aussuchen und eingeben. Dabei dürfen Sie das Stichwort bis auf zwei Zeichen abkürzen. Ist die Information länger als 23 Zeilen, stoppt die Anzeige, sobald der Bildschirm gefüllt ist und Sie kommen mit der Leertaste zum nächsten Bildschirm. Geben Sie vor Anwahl der HELP-Funktion ein Control P (^P) ein, werden alle Informationen auf dem Drucker ausgedruckt.

Wollen Sie den Text in der Datei HELP.HLP verändern, müssen Sie zuerst HELP mit der Option EXTRACT eingeben, können dann

Ihren Text einfügen und erstellen die neue, von CP/M lesbare Datei mit der Option CREATE.

Das Kommando HELP ist besonders für Anfänger sehr hilfreich, da man das Handbuch zu CP/M nicht immer parat haben muß. Auch in anderen bekannten Software-Paketen wurde diesem Beispiel von Digital Research nachgeeifert - wieviele Programme haben mittlerweile die sogenannten HELP-SCREENS integriert, die einem die Arbeit mit diesen Programmpaketen erleichtern sollen.

Besonders um CP/M 3.0+ kennenzulernen sollten Sie sich eingehender mit den HELP-Texten befassen. Sie erlernen das Bedienen von CP/M 3.0+ und lernen zusätztlich noch die meisten verfügbaren Komamndos kennen.

IX.14 HEXCOM (CP/M 3.0)

HEXCOM.COM

Erstellt eine direkt ausführbare Datei mit der Kennung COM

Eingabeformat:

HEXCOM DATEI
HEXCOM

Beschreibung:

Dieses Programm brauchen Sie, wenn Sie in Assembler programmieren. HEXCOM wandelt mit dem Programm MAC.COM erstellte Dateien im INTEL-Hex-Format in ausführbare Dateien mit der Kennung COM um.

IX.15 INITDIR (CP/M 3.0)

INITDIR.COM

Bereitet ein Disketten-Inhaltsverzeichnis darauf vor, mit Zeit- und Datumsstempel versehen zu werden

Eingabeformat:

INITDIR B:

Beschreibung:

CP/M 3.0 bietet die Möglichkeit, Dateien mit einem Zeit- und Datumsstempel zu versehen. Die entsprechenden Informationen werden in einem separaten Teil des Directorys gespeichert. Deshalb muß das Inhaltsverzeichnis jeder Diskette ent-

sprechend vorbereitet werden, indem man das Programm INITDIR aufruft. Mit dem Programm SET wählen Sie die Art der Kennzeichnung der Dateien aus.

Anwendung:

Geben Sie INITDIR am Besten bei neuen Disketten ein, da die Einträge im Inhaltsverzeichnis Platz verbrauchen. Benutzen Sie INITDIR mit einer bereits beschriebenen Diskette, machen Sie vorher eine Sicherheitskopie. Wird nämlich das Programm INITDIR während der Arbeit, zum Beispiel durch einen Stromausfall, unterbrochen, verlieren Sie sämtliche Einträge.

Sie sehen, daß es sich nicht nur bei der Systemdiskette von CP/M 3.0 lohnt, eine Sicherheitskopie anzufertigen.

IX.16 LIB (CP/M 3.0)

LIB.COM

Erstellt und verändert eine "Bibliothek" von Unterprogrammen

Eingabeformat:

LIB DATEI.XXX[Optionen]

Beschreibung:

Viele Compiler und die Assembler RMAC und MACRO-80 benutzen die Methode der separaten Unterprogramme, meistens mit der Kennung REL oder IRL versehen. Eine Zusammenstellung der wichtigsten Unterprogramme in einer "Bibliothek" können Sie mit LIB erreichen.

IX.17 LINK (CP/M 3.0)

LINK.COM

Erstellt eine ausführbare Datei aus Unterprogramm-Modulen

Eingabeformat:

LINK DATEI, DATEI2, DATEI3,..[Optionen]

Beschreibung:

Viele Compiler und die Assembler RMAC und MACRO-80 benutzen die Methode der separaten Unterprogramme, meistens mit der Kennung REL oder IRL versehen. Mit LINK können diese Dateien zu einer COM-Datei zusammengefasst werden.

IX.18 LOAD (CP/M 2.2 und CP/M 3.0)

LOAD.COM

Erstellt aus einer HEX-Datei eine COM-Datei

Eingabeformat:

LOAD Datei (ohne **Datei-Kennung!)**

Beschreibung:

Der Assembler ASM legt seinen in zwei Passes erstellten Code in Form ein Hexadezimal-Datei an. Die Form dieser Hexadezimal-Datei ist durch die Firma INTEL definiert worden. Das Programm LOAD bearbeitet nun diese HEX-Datei in der Form, daß es aus den im ASCII-Code angegebenen HEX-Daten eine ausführbare COM-Datei macht. Es ist darauf zu achten, daß der Dateityp ".HEX" nicht angegeben werden muß bzw. darf. Diese Information ist obligatorisch, da der Assembler die HEX-Datei unter dem Namen Datei.HEX abspeichert.

Anwendung:

LOAD ist wohl nach jedem Assemblierungsvorgang aufzurufen, da man nur mit dem Hexcode eines Programmes nicht allzuviel anfangen kann. Die genaue Anwendung dieses und der anderen zugehörigen Kommandos können Sie im Kapitel VII. nachlesen.

IX.19 MAC (CP/M 3.0)

MAC.COM

Ein Macroassembler für Programme, die in Assembler geschrieben sind

Eingabeformat:

MAC DATEI
MAC DATEI $Optionen

Beschreibung:

Drei Dateien werden von dem Macroassembler MAC erstellt, dazu eine Macro-Bibliothek. Das Quellprogramm hat die Kennung ASM und dic "Bibliothek" LIB. Der assemblierte Programmcode steht in der Datei mit der Kennung HEX, die Symbol-Tabelle in der SYM-Datei und das ausdruckbare Listing in der PRN-Datei. Statt mit den üblichen Klammern, werden Optionen mit dem Dollarzeichen ($) gekennzeichnet.

IX.20 MOVCPM (CP/M 2.2)

MOVCPM.COM

Verschieben des Systems in 256-Bytes-Einheiten

Eingabeformat:

MOVECPM <Faktor>*

Beschreibung:

Der Befehl MOVCPM dient dazu, das System CP/M 2.2 in 256-Bytes-Schritten zu verschieben (zu rellokieren). Der <Faktor> darf zwischen 64 und 179 liegen. Bei Angabe von 179 als <Faktor> wird ein Standard-CP/M erzeugt, entsprechend wird beispielsweise bei MOVCPM 178* ein um 256 Bytes nach unten verschobenes CP/M 2.2 erstellt. Der <Faktor> gibt den für CP/M-Kommandos (transiente) und Benutzer-Programme verfügbaren Speicherplatz in 256-Bytes-Einheiten an.

Anwendung:

Wird es aus irgendeinem Grund notwendig das System zu verschieben, beispielsweise weil sich BIOS/BDOS mit dem Anwenderprogramm überlagern, so kann man dies mittels MOVCPM erreichen. Das verschobene System kann dann mit dem SYSGEN-Kommando (siehe IX.33) abgespeichert werden.

IX.21 PATCH (CP/M 3.0)

PATCH.COM

Installiert Änderungen in CP/M 3.0 oder Programmen

Eingabeformat:

PATCH DATEI
PATCH DATEI n

Beschreibung.

Mit dem Programm PATCH können Sie Änderungen am CP/M-System vornehmen und in CP/M oder eines der transienten Programme einfügen. Diesen Prozess nennt man patchen. Mit PATCH werden die Veränderungen automatisch in das jeweilige Programm eingefügt, bei mehreren Patches bezieht man sich mit Referenznummern auf den entsprechenden Patch. Die Referenznummern beginnen mit 1.

IX.22 PIP (CP/M 2.2 und CP/M 3.0)

PIP.COM

Kopiert Dateien und transportiert Dateien zwischen Peripheriegeräten

Eingabeformat:

PIP
PIP Zieldatei=Quelldatei[Option]
PIP B:=Datei.XXX
PIP Alles.txt=TEXT1.TXT,TEXT2.TXT...
PIP AB1?*.*=AB2?*.*[Option]

Beschreibung:

PIP ist wohl das leistungsstärkste Programm bei CP/M. Sie können damit Dateien kopieren und zwar von einer Diskette zur anderen, aber auch zwischen verschiedenen Benutzerbereichen, können aus mehreren Dateien eine machen lassen, Ihre zuletzt bearbeiteten Dateien automatisch sichern, Zeilen durchnummerieren, alles in Großbuchstaben umwandeln und das achte Bit auf Null setzten lassen. Wenn Sie genauere Informationen brauchen, sehen Sie bitte in Kapitel VI nach.

Anwendung:

PIP kann mit oder ohne Optionen verwendet werden. Wenn PIP zur Ausführung von Befehlen bereit ist, zeigt es sein Bereitschaftszeichen, den Stern (*). Es erstellt die Zieldatei als genaue Kopie der Quelldatei, außer wenn bestimmte Optionen mit angegeben werden. Die eckigen Klammern müssen direkt, ohne Leerzeichen, hinter dem Dateinamen stehen. Für Buchstaben der Dateinamen, die Sie nicht wissen oder wenn Sie mehrere Dateien einer bestimmten Kategorie kopieren

wollen, können Sie auch die Zeichen "?" für unbekannte Zeichen im Namen oder "*" für alle Zeichen verwenden. Hier ein paar Beispiele:

Ihr Standardlaufwerk ist A:.

PIP B:=DATEI.XXX[V]

Mit diesem Befehl kopieren Sie die Datei DATEI.XXX von A: nach B: und lassen PIP die Korrektheit der neuen Datei überprüfen.

PIP B:=DATEI?.*[V]

So kopieren Sie eine Reihe von Dateien mit dem Namen DATEI, gleich welche Kennung sie haben, von A: nach B:.

PIP B:=*.*[V]

Damit werden alle Dateien eines Laufwerks auf das andere Laufwerk überspielt und das korrekte Kopieren überprüft.

PIP DATEINEU.XXX=DATEIALT.XXX

Mit diesem Befehl kopieren Sie die Datei DATEIALT.XXX in die neu erstellte Datei DATEINEU.XXX auf dem gleichen Laufwerk. Beide Dateien sind anschließend gleichzeitig auf dieser Diskette vorhanden.

PIP ALLES.TXT=TEXT1.TXT,TEXT2.TXT,...

So bringen Sie mehrere Dateien in einer Datei unter, die dann alle anderen enthält.

PIP B:[G] =TEXT.TXT

Die Datei TEXT.TXT wird auf das Laufwerk B: und in den Benutzerbereich "5" kopiert.

Optionen

A Archiv-Funktion. Kopiert nur Dateien, die seit der letzten Archivierung erstellt oder bearbeitet wurden. Die Funktion Zeit- und Datumstempel muß gesetzt sein.

C Confirm. CP/M fragt bei jeder Datei nach, ob diese kopiert werden darf.

Dn Lösche nach n Spalten. PIP löscht alle Zeichen, die nach der n-ten Spalte in der Datei stehen. Nur für Textdateien benutzen.

E Zeige (echo) Text auf dem Bildschirm. Der Inhalt gerade kopierter Dateien wird auf dem Bildschirm angezeigt. Nicht mit dem Parameter "N" zusammen benutzen. Nur für Textdateien einsetzen.

F Beseitige das Zeichen für "nächste Seite" (Form Feed). Einige Drucker benötigen ein (^L), um die nächste Seite anzusteuern. Wenn Ihr Drucker das nicht braucht, können Sie die entsprechenden Steuerzeichen mit "F" beseitigen.

Gn Hole oder bringe eine Datei von oder zu Benutzerbereich "n". Diese Angabe muß direkt hinter dem ersten Dateinamen stehen und ist die einzige, die dort stehen darf. Beispiel:

```
PIP B:[G5]=TEXT.TXT[G1]
```

Kopiert die Datei TEXT.TXT von Benutzerbereich "1" nach Benutzerbereich "5".

H Übertragung von hexadezimalen Daten. Diese Option sollten Sie immer eingeben, wenn Sie HEX-Dateien übertragen wollen. PIP überprüft dann den Inhalt daraufhin, ob die Daten in einwandfreiem INTEL-Format geschrieben sind.

I Ignoriere das Datei-Ende-Zeichen in HEX-Dateien. Geben Sie diese Option für jede Datei ein, außer der letzten. Mit der Option "I" wird gleichzeitig die Option "H" von PIP gesetzt. Verwenden Sie für die letzte Datei den Parameter "H":

```
PIP DATEI.HEX=PROG1.HEX[I],PROG2.HEX[H]
```

K Unterdrücke die Anzeige der Dateinamen während des Kopierens.

L Übersetze alle Großbuchstaben in Kleinbuchstaben. Benutzen Sie diese Option nur für Text-Dateien und verwenden Sie zusätzlich den Parameter "Z" für WordStar-Dateien.

N Nummeriert die Zeilen einer Datei fortlaufend. Die Zahlen beginnen mit 1 in der ersten Zeile und werden pro Zeile und eins erhöht. Die Ziffern können maximal sechstellig sein, nicht gebrauchte Stellen erscheinen als Leerzeichen. Ein Doppelpunkt und ein Leerzeichen stehen hinter den Ziffern. Nicht mit den Optionen "E" oder "N" zusammen benutzen. Für WordStar-Dateien die Optionen "Z" mit eingeben.

N2 Nummeriert die Zeilen für ein BASIC-Programm.

O Übertragung von Objekt-Dateien. Wird benutzt, um
Nicht-Text- und Nicht-COM-Dateien zu übertragen.
Nicht für Text-Dateien verwenden.

Pn Setzt die Seitenlänge. Voreinstellung ist 60
Zeilen pro Seite. Die "F"-Option sollten Sie mit
eingeben, damit vorhandene Zeichen für das
Seitenende (^L) gelöscht werden. Nur für Text-
Dateien verwenden.

Qxxxx^Z Ende des Kopierens nach dieser Zeichenkette. PIP
kopiert eine Datei bis und einschließlich der
eingegebenen Zeichenkette. Benutzten Sie den
Befehl mit zwei Befehlszeilen, damit die Zeichen-
kette nicht in Großbuchstaben übersetzt wird:

```
PIP
CON:=TEXT.TXT[Weiler]
```

Schreiben Sie alles in die erste Zeile hinter
PIP, erscheint die Eingabe in Großbuchstaben.

R Kopiere SYSTEM-Dateien. Diese Dateien werden mit
DIR nicht angezeigt und von PIP normalerweise
nicht kopiert. Mit "R" kann diese Einschränkung
aufgehoben werden.

Sxxxx^Z Beginne bei dieser Zeichenkette. PIP beginnt mit
dem Kopieren an dieser Zeichenkette. Schreiben
Sie den Befehl in zwei Zeilen, sonst wird die
Zeichenkette in Großbuchstaben übersetzt. Nur für
Text-Dateien verwenden.

Tn Schrittweite des Tabulators einstellen. Normaler-
weise arbeitet CP/M mit einer Schrittweite von
acht Zeichen. Beim Drucken einer Datei funktio-

niert dies jedoch nicht und die Schrittweite muß definiert werden. "n" gibt die Anzahl der Leerzeichen für einen Tabulator-Schritt ein. Nur für Text-Dateien benutzen.

U Wandle in Großbuchstaben um. Wandelt alle Kleinbuchstaben in Großbuchstaben um. Nur mit Text-Dateien verwenden. Für WordStar-Dateien zusätzlich die "Z"-Option setzen.

V Überprüfen. Dieser Parameter sorgt dafür, daß PIP die kopierte Datei genau mit der Ursprungsdatei vergleicht.

W Dateien mit dem Attribut Read Only (RO) überschreiben. Diese Dateien können normalerweise nur gelesen, aber nicht verändert oder gelöscht werden. Mit der "W"-Option werden diese Dateien ohne Rückfrage gelöscht.

Z Das Paritäts-Bit (8.Bit) löschen. Der ASCII-Zeichensatz benutzt nur sieben Bits. Das achte Bit wird von verschiedenen Programmen für verschiedene Aufgabe eingesetzt. Die "Z"-Option ist nicht notwendig beim Kopieren nach ASCII-Einheiten, wie CON: oder LST:.

IX.23 PUT (CP/M 3.0)

PUT.COM

Schreibt die Daten für den Drucker oder den Bildschirm in eine Datei

Eingabeformat:

PUT CONSOLE OUTPUT TO FILE DATEI.XXX[OPTION]
PUT PRINTER OUTPUT TO FILE DATEI.XXX[OPTION]
PUT CONSOLE OUTPUT TO CONSOLE
PUT PRINTER OUTPUT TO PRINTER

Beschreibung:

Normalerweise schickt CP/M die Daten für den Bildschirm zum Bildschirm und die Daten für den Drucker zum Drucker. Mit dem PUT-Befehl können die Daten zusätzlich in eine Datei geschrieben werden.

Anwendung:

Die ersten beiden Zeilen oben sagen CP/M, es soll eine Datei mit dem Namen DATEI.XXX eröffnen und alles, was für den Bildschirm oder den Drucker bestimmt ist, dort hineinschreiben. Ist die Operation beendet, kehrt das Programm wieder zum normalen CP/M-Modus zurück. Die beiden letzten Zeilen brechen das PUT-Programm ab. Diese können Sie verwenden, wenn Sie beispielsweise eine SUBMIT-Datei haben, die den PUT-Befehl benutzt und anschließend wieder in den Normal-Modus finden soll.

Mit der Option NO ECHO verhindern Sie die Anzeige der übertragenen Daten auf dem Bildschirm. Mit der Option FILTER verwandeln Sie jeden Control Charakter in ein druckbares Zeichen.

Mit der Option SYSTEM gehen nicht nur die Daten, sondern auch die Kommandozeile in die Datei.

IX.24 REN(AME) (CP/M 2.2 und CP/M 3.0)

RENAME eingebauter Befehl oder RENAME.COM

Eine Datei umbenennen

Eingabeformat:

RENAME
RENAME DATEI2.XXX=DATEI1.XXX
RENAME AB?*2=AB?*1

Beschreibung:

Mit RENAME ändern Sie den Namen einer Datei - und nur den Namen. Der Inhalt wird nicht verändert.

Anwendung:

Wenn Die Datei nicht auf dem Standard-Laufwerk ist, kann die Laufwerksbezeichnung mit eingegeben werden. Geben Sie RENAME ohne Optionen ein, fragt das Programm danach.

IX.25 RMAC (CP/M 3.0)

RMAC.COM

Macroassembler für die Programmierung in Assembler

Eingabeformat:

RMAC DATEI
RMAC DATEI $OPTIONEN

Beschreibung:

Drei Dateien werden von dem Macroassembler RMAC erstellt, dazu eine Macro-Bibliothek. Das Quellprogramm hat die Kennung ASM und die "Bibliothek" LIB. Der assemblierte Programmcode steht in der Datei mit der Kennung REL, die Symbol-Tabelle in der SYM-Datei und das ausdruckbare Listing in der PRN-Datei. Mit dem Programm LINK wird ein ausführbares Programm aus der REL-Datei erstellt.

Anwendung:

Statt mit den üblichen Klammern, werden Optionen mit dem Dollarzeichen ($) gekennzeichnet. Vor dem Dollarzeichen muß ein Leerzeichen stehen.

IX.26 SAVE (CP/M 2.2 und CP/M 3.0)

SAVE.COM

Speichert den Inhalt des Arbeitsspeichers in eine Datei

Eingabeformat:

SAVE

Beschreibung:

Der SAVE-Befehl erstellt eine Datei mit dem Inhalt eines bestimmten Speicherbereiches. Vor dem Abspeichern muß ein anderes Programm etwas in den Speicher hineingeschrieben haben.

Anwendung:

Sie müssen SAVE aufrufen, BEVOR Sie ein anderes Programm in den Arbeitsspeicher laden. SAVE plaziert sich automatisch am obersten Ende von CP/M und übergibt dann die Kontrolle wieder an CP/M. Dann lassen Sie das Programm laufen. Sobald es beendet ist, übernimmt SAVE automatisch wieder die Kontrolle. Sie werden dann nach dem Dateinamen, der Start- und Endadresse innerhalb des Speichers gefragt.

IX.27 SET (CP/M 3.0)

SET.COM

Setzt Datei-Attribute, ermöglicht die Eingabe eines Schutzwortes, vergibt Namen für Disketten und wählt die Art der Zeit- und Datumsstempel

Eingabeformat:

SET DATEI.XXX[Option]
SET AB?*.*[Option]
SET B:[Option]
SET[Option]

Beschreibung:

Das SET-Programm wird für mehrere Dienste eingesetzt. Zu den wichtigsten Diensten gehören: Das setzten des Archiv-Modus und die Möglichkeit, Dateien und ganze Laufwerke zu Read Only-Dateien oder -Laufwerken zu erklären.

Für jede Diskette kann ein eigener Name eingegeben werden, zusätzlich noch ein Schutzwort für jede Diskette und jede Datei. Die Disketten-Namen können mit SHOW angezeigt werden.

Als dritte Möglichkeit können Sie die Dateien mit Stemplen für Zeit und Datum versehen.

Lesen Sie Kapitel V , wenn Sie nähere Informationen brauchen.

IX.28 SETDEF (CP/M 3.0)

SETDEF.COM

Zeigt oder definiert den Suchpfad und schaltet das bildschirmweise Anzeigen aus oder ein

Eingabeformat:

SETDEF
SETDEF B:
SETDEF[Option]

Beschreibung:

Normalerweise starten Sie ein transientes Programm, indem Sie den Namen eintippen und RETURN hinterher. Wollen Sie das Programm von einem anderen Laufwerk aus aufrufen, geben Sie die Laufwerksbezeichnung mit ein.

Arbeiten Sie zum Beispiel immer auf dem Laufwerk B:, aber Ihre CP/M-Dateien sind auf Laufwerk A:, können Sie CP/M vorschreiben, wo es zuerst nach den Dateien suchen soll. Das nennt man einen Suchpfad vorgeben. Damit könnten Sie CP/M sagen, daß es zuerst auf Laufwerk A:, dann auf Laufwerk C: und erst dann auf dem Standard-Laufwerk suchen soll.

IX.29 SHOW (CP/M 3.0)

SHOW.COM

Zeigt die Optionen einer Diskette an

Eingabeformat:

SHOW[Option]
SHOW B:[Option]

Beschreibung:

Mit SHOW können Sie sich sozusagen die "technischen Daten" einer Diskette anzeigen lassen. Als Information können Sie bekommen: den freien Speicherplatz auf der Diskette, die Anzahl der freien Einträge ins Inhaltsverzeichnis, die Zahl der aktiven Benutzer-Bereiche und die angewählte Benutzer-Bereichs-Nummer. Dazu noch Angaben zu der Gesamt-Speicherkapazität der Diskette, Blockgröße, Sektorengröße und Anzahl der Sektoren pro Track. Der Name einer Diskette kann auch angezeigt werden.

Beachten Sie bei der Verwendung von CP/M 2.2, daß auch dieses Kommando mittels STAT realisiert wird.

IX.30 SID (CP/M 3.0)

SID.COM

Ein Debugger-Programm, um Assembler-Programm zu laden, verändern und zu testen

Eingabeformat:

SID
SID DATEI.XXX
SID DATEI.XXX TEXT.SYM

Beschreibung:

Mit SID können Sie COM- oder HEX-Dateien in den Arbeitsspeicher laden, dort ansehen und verändern. Wenn Sie eine SYM-Datei mitladen, können Sie das Programm unter genauer Kontrolle laufen lassen. SID kann auch ablauffähige Programme in INTEL 8080 Mnemonics umwandeln.

IX.31 SUBMIT (CP/M 2.2 und CP/M 3.0)

SUBMIT.COM

Ermöglicht die Befehlseingabe aus einer Datei, statt von der Tastatur

Eingabeformat:

SUBMIT
SUBMIT DATEI.SUB
SUBMIT DATEI DATEI1 DATEI 2 DATEI3...

Beschreibung:

Befehle, die Sie sonst über die Tastatur eingeben, können sie auch in eine SUBMIT-Datei schreiben und dann automatisch abarbeiten lassen. Steht kein Befehl mehr in der Datei, kommt die Kontrolle an CP/M zurück und Sie können normal weitermachen.

CP/M sucht bei jedem Neustart eine Datei namens PROFILE.SUB und führt die dort stehenden Befehle aus, wenn es diese Datei findet.

Anwendung:

Sie schreiben eine SUBMIT-Datei einfach mit Ihrem Textprogramm, indem Sie die gewünschten Befehle jeweils in eine Zeile schreiben. Als Kennung muß diese Datei SUB haben, sonst wird sie von SUBMIT nicht anerkannt.

STAT (CP/M 2.2)

STAT.COM

Gibt die Speicherbelegung und den Schreibschutz- und Systemzustand der angegebenen Dateien in einer Liste wieder. Ferner gibt STAT den Zustand aller im System derzeit aktivierten Laufwerke in einer Liste wieder.

Eingabeformat:

STAT
STAT Datei.XXX
STAT Dat??.*
STAT DEV:
STAT DSK:
STAT USR:
STAT Datei.XXX $SYS

Beschreibung:

Mit STAT können SIe sich praktisch die Optionen Ihrer Disketten und der darauf befindlichen Files ausgeben lassen. Das Kommando ist also ähnlich dem CP/M 3.0-Kommando SHOW. Sie können sich den freien Speicher auf Ihrer Diskette ausgeben lassen, die Länge bestimmter beliebieger Files in Records oder beispielsweise die benutzen User-Bereiche. Also ist das STAT-Kommando noch das CP/M 3.0-Kommando DEVICE in einem. Mit STAT DSK: können Sie sich alle Charakteristika einer beliebigen Diskette ausgeben lassen. Ferner können Sie mittels des STAT-Kommandos Dateien auf $R/O (Read Only), $R/W (Read and Write), $SYS (System-Datei) und $DIR (Directory-File) setzen. Dieser Status wird auch mittels des STAt-Kommandos angezeigt. STAT beinhaltet also auch noch das CP/M 3.0-Kommando SET. Sie sehen, STAT ist sehr flexibel und umfangreich.

STAT im einzelnen:

STAT gibt den Zustand aller im System derzeit aktiven Laufwerke in einer Liste wie folgt aus:

```
<Laufwerk>: <Schutz>, Space: <n>k
```

STAT USR: gibt die aktuelle Benutzer-Nummer sowie alle auf der Diskette belegten Benutzer-Bereiche aus.

```
Active User : 5
Active Files: 0 1 3 5 7
```

STAT DEV: entspricht dem CP/M-3.0-Kommando DEVICE und gibt eine Liste der Ein-/Ausgabegeräte aus. Eine Neuzuordnung ist möglich.

```
CON: is CRT:
RDR: is TTY:
PUN: is TTY:
LST: is LPT:
```

STAT <Kanal>=<Gerät> ordnet einem Kanal ein neues Ein-/Ausgabegerät zu.

```
STAT CON:=CRT:, LST:=TTY:
```

STAT <Laufwerk> $R/O bzw. STAT <Laufwerk> $R/W setzt das Laufwerk> setzt das angegebene Laufwerk auf Nur-Lesen bzw. Schreib/Lese-Status.

STAT <Laufwerk>: gibt den auf der angegebenen Diskette derzeit verfügbaren Speicherplatz aus.

```
Bytes Remaining On A: 5k
```

STAT <Dateiname.XXX> evtl. mit Jokern gibt die Speicherbelegung und den Schreibschutz- bzw. Systemzustand des/der angegebenen Dateien aus.

```
Recs Bytes EXT Acc
 144 18k 2 R/W A:(D2.BAS)
  14 2k 1 R/W A:DISC.BAS
```

Ist eine Datei eingeklammert, so handelt es sich hier um eine System-Datei, also eine Datei, die mit der $SYS-Option versehen wurde. Sie erscheint nicht im "normalen" Inhalts-Verzeichnis.

Durch die Option $S können Sie sich noch zusätzlich die Größe (Size) einer Datei ausgeben lassen, die aber meistens mit der Anzahl der Records zusammenfällt.

STAT <Dateiname.XXX> $<Option> setzt den oder die Dateien (wenn mit Joker) auf den Status <Option>. Folgende Optionen sind möglich:

```
$R/O - (Read-Only) Schreibschutz setzen
$R/W - (Read-Write) Schreibschutz aufheben
$SYS - (System) Systemeigenschaft setzen
$DIR - (Directory) Systemeigenschaft löschen
```

STAT <Laufwerk>:DSK: listet die in einer im BIOS-Teil angegebene Tabelle wie folgt aus:

```
   A: Drive Characteristics
1368: 128 Byte Record Capacity
 171: Kilobyte Drive Capacity
  64: 32 Byte Directory Entries
  64: Checked Directory Entries
 128: Records/ Extent
```

```
  8: Records/ Block
 36: Sectors/ Track
  2: Reserved Tracks
```

STAT VAL: gibt eine Liste der mit STAT möglichen Kommandos aus. Diese Liste sieht wie folgt aus:

```
Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status : DSK: d:DSK:
User Status : USR:
IObyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

Anwendung:

Das STAT-Kommando gibt ein wenig mehr Informationen über die Dateigröße und -verteilung, als das Kommando DIR dies tut. Ferner kann man sich die aktuelle Belegung der Ein-/Ausgabegeräte (DEVices) ausgeben lassen.

Da das STAT-Kommando die CP/M 3.0-Kommandos SET, DEVICE, SHOW und teilweise DIR mit Optionen beinhaltet, können Sie die Bedeutung auch bei diesen Kommandos nachschlagen.

SYSGEN (CP/M 2.2)

SYSGEN.COM

Kopiert das System CP/M 2.2 auf Diskette

Eingabeformat:

SYSGEN
SYSGEN Datei.XXX
SYSGEN *

Beschreibung:

SYSGEN schreibt das Ergebnis eines MOVCPM-Kommandos (s.u.) auf die System-Spur einer Diskette. SYSGEN * schreibt dabei das unmittelbar zuvor durch das MOVCPM-Kommando erstellte System auf Diskette. SYSGEN <Dateiname> sichert eine durch MOVCPM erstellte Systemdatei auf die System-Spuren einer Diskette. Schließlich SYSGEN ohne Parameter fragt nach Quell- und Zieldiskette und kopiert dann dieses System.

Mit MOVCPM kann man CP/M 2.2 im Speicher verschieben, manchmal ist dies notwendig. Sie können CP/M in Schritten von 256 Bytes verschieben. Die Position im Speicher wird durch einen Wert zwischen 64 und 179 angegeben. Die Syntax lautet:

MOVCPM <Größe>* als Beispiel: MOVCPM 178*

MOVCPM 178* erzeugt eine gegenüber dem Standard-CP/M um 256 Bytes nach "unten" verschobenes CP/M. Sie können dieses verschobene System nun auf Diskette sichern oder mittels SYSGEN direkt als System abspeichern.

IX.34 TYPE (CP/M 2.2 und CP/M 3.0)

TYPE eingebaut und TYPE.COM

Zeigt eine Text-Datei auf dem Bildschirm

Eingabeformat:

TYPE
TYPE DATEI.XXX
TYPE AB?.*
TYPE DATEI.XXX[NO PAGE]
TYPE AB?.*[NO PAGE]

Beschreibung:

Der TYPE-Befehl listet eine oder mehrere Dateien auf dem Bildschirm. Sobald der Bildschirm voll ist, stoppt die Anzeige und wird mit der Leertaste oder RETURN wieder in Gang gesetzt. Geben Sie die Option "NO PAGE" mit an, stoppt die Bildschirmausgabe nicht.

Anwendung:

Die eingebaute Version von TYPE können sie von jedem Laufwerk und aus jedem Benutzerbereich heraus aufrufen. Wenn Sie das Programm abbrechen wollen, geben Sie Control C (^C) ein. Sie erhalten einen Ausdruck Ihrer Datei, wenn Sie noch vor dem RETURN den Befehl Control P (^P) eingeben.

IX.35 USER (CP/M 2.2 und CP/M 3.0)

USER eingebauter Befehl

Verändert den aktuellen Benutzerbereich

Eingabeformat:

USER
USER n

Beschreibung:

Jede Diskette kann in 16 unterschiedliche Benutzerbereiche unterteilt werden. So hat man für unterschiedliche Arbeiten jeweils einen Benutzerbereich, in dem nur die notwendigen Dateien und Programme stehen. SYSTEM-Dateien in der Benutzerebene Null, können aus allen Benutzerbereichen heraus aufgerufen werden.

Anwendung:

Der aktuelle Benutzerbereich wird im CP/M-Prompt angezeigt (1B>) und kann durch die Eingabe von USER verändert werden. Geben Sie den neuen Bereich nicht mit an, fragt das Programm danach. Mit dem Programm SHOW können Sie sich die aktiven Benutzerbereiche anzeigen lassen.

IX.36 XREF (CP/M 3.0)

XREF.COM

Erstellt eine Referenzliste für Assembler-Programme

Eingabeformat:

XREF DATEI
XREF DATEI $P

Beschreibung:

Die Assembler MAC und RMAC erstellen eine alphabetische Liste aller Symbole und deren Werte in einem Programm. XREF erhöht den Komfort dadurch, daß es zu jedem Symbol noch mit angibt, in welcher Programmzeile es zu finden ist. Das Programm XREF benötigt die Dateien SYM und PRN.

X. Unterschiede der CPC-Rechner

Die Firma Schneider, die mit den Firmen Amstrad und Locomotive Software zusammenarbeitet, hat wohl in der letzten Zeit für die meiste Aufregung auf dem deutschen Markt gesorgt. Immerhin hat diese Firma es fertig gebracht, innerhalb nicht ganz eines Jahres 3 (!) Rechner auf den Markt zu bringen, die zwar kompatibel sein soll(t)en, aber trotzdem einige wichtige Veränderungen aufzuweisen haben.

Der erste Rechner dieser Reihe war der Schneider CPC 464. Dieser hat ein eingebautes Kassettenlaufwerk und war bzw. ist wegen seines schnellen und komfortablen BASICs sowie des günstigen Preises sehr beliebt. Es dauerte nicht allzu lange, (es war während der Hannover-Messe 1985), da kündigte die Firma Schneider ein neues Gerät an: Den CPC 664.

Der wohl wichtigste Unterschied zum CPC 464 sollte das eingebaute Diskettenlaufwerk sein. Beide Geräte verfügen über 64 KBytes RAM-Speicher, von dem allerdings eine Menge für Bildschirm (16 KBytes) sowie Systemvariablen verloren geht. Der Benutzer kann unter BASIC über 42249 Bytes frei verfügen. Der CPC 664 hat ein leicht verändertes BASIC (V 1.1), das einige Fehler des CPC-464-BASIC korrigiert, sowie einige neue Kommandos. Für beide Rechner, CPC 464 und CPC 664, wird CP/M 2.2 benötigt. CP/M 2.2 ist die gängige CP/M-Version für 64-KBytes-Rechner.

Aber nicht nur das Innenleben hat sich geändert, auch das äußere Erscheinungsbild hat einen Wandel vollzogen: Der CPC 664 ist etwas flacher und verfügt über eine andere Tastatur mit hervorgehobenen Cursor-Block. Der CPC ist professioneller geworden. Allerdings haben die Software-Häuser so Ihre Probleme mit den zwei "Brüdern", denn leider ist es nicht so weit her mit der Kompatibilität. Das ist ja klar: Wenn man

Fehler aus dem BASIC eliminiert und Kommandos hinzufügt, so können die zwei Rechner ja nun nicht kompatibel sein.

Viel schlimmer ist sind allerdings die Maschinensprache-Programmierer betroffen, da sich hier am meisten verändert hat. Doch gottlob: es gibt ja CP/M. Hier hat sich für den Benutzer überhaupt nichts getan. Sie sehen, wie toll es sein kann, über einen Standard zu verfügen.

Da die Konkurrenzfirmen Computer auf den Markt gebracht haben, die über 128 und mehr KBytes verfügen, sah sich die Firma Schneider mit ihren Partnern wohl gezwungen, auch einen 128-KBytes-Rechner rauszubringen. Und siehe da, Schneider überraschte mit dem Schneider CPC 6128.

Auch der CPC 6128 verfügt über ein eingebautes Floppy-Laufwerk, darum ist wohl die erste Ziffer in der Typenbezeichnung auch eine "6". Der CPC 6128 ähnelt dem CPC 664 auch mehr als dem CPC 464. Der 128-KBytes-Rechner ist noch flacher, man ist schon geneigt zu sagen *superflach*. Er verfügt über eine fast noch professionellere Tastatur als der CPC 664. Vor allem die Cursor-Tasten mußten wieder mal dran glauben; man hat sie jetzt ganz wo anders hin plaziert. Allerdings ist die Tastatur ein wenig gewöhnungsbedürftig: Die COPY-Taste ist nicht mehr in der Mitte des Cursor-Blocks und die Shift-Tasten befinden sich nicht in der untersten Reihe sondern sind über die Tasten "CONTROL" bzw. "ENTER" gelagert. Hier vertippt man sich zumindest anfangs schon dann und wann.

Das BASIC hat sich in der CPC 6128-Version nicht verändert, es ist immer noch die Version 1.1 eingebaut. Das AMSDOS, also das eingebaute Betriebssystem für die Disketten-Stationen hat sich erfreulicherrweise in allen drei Rechnern nicht verändert. Übrigens sind die Drei-Zoll-Laufwerke sehr schnelle und zuverlässige Laufwerke, da muß sich manch ein anderes Laufwerk erst einmal eine Scheibe von abschneiden.

Ferner haben sich die Anschlüße an der Hinterseite des Rechners ein wenig verändert. Floppy-Disc-Anschluß und Drucker-Anschluß (Centronics) sind qualitativ gestiegen. Zusätzlich verfügt der CPC 6128 noch über einen Anschluß mit der Bezeichnung *Extension.*

Beim Kauf der DDI-1 bzw. des CPC 664 erhalten Sie eine System-Diskette, auf der sich das CP/M 2.2 sowie die Programmiersprache LOGO befinden. Beim CPC 6128 werden sogar zwei Disketten ausgeliefert, da man nicht alle Dateien des CP/M 3.0 auf einer Diskette unterbringen konnte. Vor allem die HELP-Texte, also die Texte, die Sie mittels des CP/M-3.0-Kommandos HELP abrufen können, brauchen eine Menge Platz auf der Diskette. Ferner erhält der CPC 6128-Käufer auch eine Diskette mit CP/M 2.2

Der gravierenste Unterschied des neuen CPC zu den Vorgängern ist allerdings der verfügbare Speicherplatz. Er hat sich hier direkt verdoppelt. Allerdings hat der BASIC-Programmierer hiervon weniger; er verfügt weiterhin lediglich über 42249 Bytes. Ohne Tricks ist hier nichts zu machen. Doch die 128-KBytes-verfügbarer Speicher im CPC 6128 verleihen dem Rechner die Möglichkeit des Sprunges vom CP/M 2.2- zum CP/M 3.0+-Rechner. CP/M 3.0 ist die gängige CP/M-Version für 128-KBytes-Rechner. Das Anhängsel + steht hier für einige Änderungen bzw. Ergänzungen des CP/M 3.0.

Mit beiden CP/M-Versionen läßt es sich eigentlich sehr angenehm arbeiten. Natürlich ist das CP/M 3.0+ etwas komfortabler, aber das liegt in der Natur der Sache; auch im neuen Golf fährt es sich etwas bequemer als in einem alten Käfer. Dazu sind Fortschritte schließlich da und die Firma Digital Research plfegt ihr Nobel-Programm CP/M immer weiter.

Da dieses Buch als Trainingsbuch für alle CPC-Rechner dienen soll, werden sowohl die CP/M-2.2- als auch die CP/M-3.0-Kommandos beschrieben. Fast alle CP/M-2.2-Kommandos sind auch unter CP/M 3.0 zu verwenden. Befehle, die Sie nur unter CP/M 3.0 verwenden können, sind entsprechend markiert.

Anhang 1

Umrechnungstabelle

Umrechnungstabelle

DEZIMAL	HEXA-DEZIMAL	BINÄR	ASCII
0	00	000 0000	NUL
1	01	000 0001	SOH
2	02	000 0010	STX
3	03	000 0011	ETX
4	04	000 0100	EOT
5	05	000 0101	ENQ
6	06	000 0110	ACU
7	07	000 0111	BEL
8	08	000 1000	ES
9	09	000 1001	HT
10	0A	000 1010	LF
11	0B	000 1011	VT
12	0C	000 1100	FF
13	0D	000 1101	CR
14	0E	000 1110	SO
15	0F	000 1111	SI
16	10	001 0000	DLE
17	11	001 0001	DC1
18	12	001 0010	DC2
19	13	001 0011	DC3
20	14	001 0100	DC4
21	15	001 0101	AAK
22	16	001 0110	SYU
23	17	001 0111	ETB
24	18	001 1000	CAN
25	19	001 1001	EM
26	1A	001 1010	SUB
27	1B	001 1011	ESC
28	1C	001 1100	FS
29	1D	001 1101	GS
30	1E	001 1110	RS

DEZIMAL	HEXA-DEZIMAL	BINÄR	ASCII
31	1F	001 1111	VS
32	20	010 0000	SP
33	21	010 0001	!
34	22	010 0010	"
35	23	010 0011	#
36	24	010 0100	$
37	25	010 0101	%
38	26	010 0110	&
39	27	010 0111	'
40	28	010 1000	(
41	29	010 1001	)
42	2A	010 1010	*
43	2B	010 1011	+
44	2C	010 1100	,
45	2D	010 1101	-
46	2E	010 1110	.
47	2F	010 1111	/
48	30	011 0000	0
49	31	011 0001	1
50	32	011 0010	2
51	33	011 0011	3
52	34	011 0100	4
53	35	011 0101	5
54	36	011 0110	6
55	37	011 0111	7
56	38	011 1000	8
57	39	011 1001	9
58	3A	011 1010	:
59	3B	011 1011	;
60	3C	011 1100	<
61	3D	011 1101	=
62	3E	011 1110	>
63	3F	011 1111	?

DEZIMAL	HEXA-DEZIMAL	BINÄR	ASCII
64	40	100 0000	@
65	41	100 0001	A
66	42	100 0010	B
67	43	100 0011	C
68	44	100 0100	D
69	45	100 0101	E
70	46	100 0110	F
71	47	100 0111	G
72	48	100 1000	H
73	49	100 1001	I
74	4A	100 1010	J
75	4B	100 1011	K
76	4C	100 1100	L
77	4D	100 1101	M
78	4E	100 1110	N
79	4F	100 1111	O
80	50	101 0000	P
81	51	101 0001	Q
82	52	101 0010	R
83	53	101 0011	S
84	54	101 0100	T
85	55	101 0101	U
86	56	101 0110	V
87	57	101 0111	W
88	58	101 1000	X
89	59	101 1001	Y
90	5A	101 1010	Z
91	5B	101 1011	Ä
92	5C	101 1100	Ö
93	5D	101 1101	Ü
94	5E	101 1110	^
95	5F	101 1111	_
96	60	110 0000	`

DEZIMAL	HEXA- DEZIMAL	BINÄR	ASCII
97	61	110 0001	a
98	62	110 0010	b
99	63	110 0011	c
100	64	110 0100	d
101	65	110 1101	e
102	66	110 0110	f
103	67	110 0111	g
104	68	110 1000	h
105	69	110 1001	i
106	6A	110 1010	j
107	6B	110 1011	k
108	6C	110 1100	l
109	6D	110 1101	m
110	6E	110 1110	n
111	6F	110 1111	o
112	70	111 0000	p
113	71	111 0001	q
114	72	111 0010	r
115	73	111 0011	s
116	74	111 0100	t
117	75	111 0101	u
118	76	111 0110	v
119	77	111 0111	w
120	78	111 1000	x
121	79	111 1001	y
122	7A	111 1010	z
123	7B	111 1011	ä
124	7C	111 1100	ö
125	7D	111 1101	ü
126	7E	111 1110	~
127	7F	111 1111	DEL

Anhang 2

Die CP/M Control-Zeichen

Die CP/M Control-Zeichen

Befehl ======	Wirkung =======
^A	Cursor ein Zeichen nach links*
^B	Cursor an Anfang der Zeile oder an's Ende, wenn Cursor schon am Anfang steht
^C	CP/M abbrechen; Reset durchführen
^E	Cursor zur nächsten Zeile
^F	Cursor ein Zeichen rechts*
^G	Zeichen unter Cursor löschen*
^H	Lösche Zeichen links vom Cursor
^I	Cursor zur nächsten TAB-Position
^J	Ausführungsbefehl (Zeilenschaltung)
^K	Lösche von Cursor bis Zeilenende
^M	wie RETURN
^P	Drucker ein/ausschalten
^Q	Scrollen einschalten nach ^S
^R	Befehlszeile wiederholen
^S	Bildschirm-Ausgabe anhalten
^U	Alle Zeichen in Zeile löschen

^W	Alte Befehlszeile wieder anzeigen*
^X	Zeichen links vom Cursor löschen
^Z	String-Ende in PIP und ED

* bedeutet: nur bei CP/M mit "bankswitching" verfügbar

Anhang 3

Alle PIP-Parameter

PIP - Parameter

A — Archiv-Funktion. Kopiert nur Dateien, die seit der letzten Archivierung erstellt oder bearbeitet wurden. Die Funktion Zeit- und Datumsstempel muß gesetzt sein.

C — Confirm. CP/M fragt bei jeder Datei nach, ob diese kopiert werden darf.

Dn — Lösche nach n Spalten. PIP löscht alle Zeichen, die nach der n-ten Spalte in der Datei stehen. Nur für Textdateien benutzen.

E — Zeige (echo) Text auf dem Bildschirm. Der Inhalt gerade kopierter Dateien wird auf dem Bildschirm angezeigt. Nicht mit dem Parameter "N" zusammen benutzen. Nur für Textdateien einsetzen.

F — Beseitige das Zeichen für "nächste Seite" (Form Feed). Einige Drucker benötigen ein (^L), um die nächste Seite anzusteuern. Wenn Ihr Drucker das nicht braucht, können Sie die entsprechenden Steuerzeichen mit "F" beseitigen.

Gn — Hole oder bringe eine Datei von oder zu Benutzerbereich "n". Diese Angabe muß direkt hinter dem ersten Dateinamen stehen und ist die einzige, die dort stehen darf. Beispiel:

```
PIP B:[G5]=TEXT.TXT[G1]
```

Kopiert die Datei TEXT.TXT von Benutzerbereich "1" nach Benutzerbereich "5".

H — Übertragung von hexadezimalen Daten. Diesen Parameter sollten Sie immer eingeben, wenn Sie HEX-Dateien

übertragen wollen. PIP überprüft dann den Inhalt daraufhin, ob die Daten in einwandfreiem INTEL-Format geschrieben sind.

I — Ignoriere das Datei-Ende-Zeichen in HEX-Dateien. Geben Sie diesen Parameter für jede Datei ein, außer der letzten. Mit der Option "I" wird gleichzeitig die Option "H" von PIP gesetzt. Verwenden Sie für die letzte Datei den Parameter "H":

```
PIP DATEI.HEX=PROG1.HEX[I],PROG2.HEX[I],PROG3.HEX[H]
```

K — Unterdrücke die Anzeige der Dateinamen während des Kopierens.

L — Übersetze alle Großbuchstaben in Kleinbuchstaben. Benutzen Sie diesen Parameter nur für Text-Dateien und verwenden Sie zusätzlich den Parameter "Z" für WordStar-Dateien.

N — Nummeriert die Zeilen einer Datei fortlaufend. Die Zahlen beginnen mit 1 in der ersten Zeile und werden pro Zeile und eins erhöht. Die Ziffern können maximal sechstellig sein, nicht gebrauchte Stellen erscheinen als Leerzeichen. Ein Doppelpunkt und ein Leerzeichen stehen hinter den Ziffern. Nicht mit den Parametern "E" oder "N" zusammen benutzen. Für WordStar-Dateien den Parameter "Z" mit eingeben.

N2 — Nummeriert die Zeilen für ein BASIC-Programm.

O — Übertragung von Objekt-Dateien. Wird benutzt, um Nicht-Text- und Nicht-COM-Dateien zu übertragen. Nicht für Text-Dateien verwenden.

Pn — Setzt die Seitenlänge. Voreinstellung ist 60 Zeilen

pro Seite. Den "F"-Parameter sollten Sie mit eingeben, damit vorhandene Zeichen für das Seitenende (^L) gelöscht werden. Nur für Text-Dateien verwenden.

Qxxxx^Z Ende des Kopierens nach dieser Zeichenkette. PIP kopiert eine Datei bis und einschließlich der eingegebenen Zeichenkette. Benutzten Sie den Befehl mit zwei Befehlszeilen, damit die Zeichenkette nicht in Großbuchstaben übersetzt wird:

```
PIP
CON:=TEXT.TXT[Weiler]
```

Schreiben Sie alles in die erste Zeile hinter PIP, erscheint die Eingabe in Großbuchstaben.

R Kopiere SYSTEM-Dateien. Diese Dateien werden mit DIR nicht angezeigt und von PIP normalerweise nicht kopiert. Mit "R" kann diese Einschränkung aufgehoben werden.

Sxxxx^Z Beginne bei dieser Zeichenkette. PIP beginnt mit dem Kopieren an dieser Zeichenkette. Schreiben Sie den Befehl in zwei Zeilen, sonst wird die Zeichenkette in Großbuchstaben übersetzt. Nur für Text-Dateien verwenden.

Tn Schrittweite des Tabulators einstellen. Normalerweise arbeitet CP/M mit einer Schrittweite von acht Zeichen. Beim Drucken einer Datei funktioniert dies jedoch nicht und die Schrittweite muß definiert werden. "n" gibt die Anzahl der Leerzeichen für einen Tabulator-Schritt ein. Nur für Text-Dateien benutzen.

U Wandle in Großbuchstaben um. Wandelt alle Kleinbuchstaben in Großbuchstaben um. Nur mit Text-

Dateien verwenden. Für WordStar-Dateien zusätzlich den "Z"-Parameter setzen.

V Überprüfen. Dieser Parameter sorgt dafür, daß PIP die kopierte Datei genau mit der Ursprungsdatei vergleicht.

W Datein mit dem Attribut Read Only (RO) überschreiben. Diese Dateien können normalerweise nur gelesen, aber nicht verändert oder gelöscht werden. Mit dem "W"-Parameter werden diese Dateien ohne Rückfrage gelöscht.

Z Das Paritäts-Bit (8.Bit) löschen. Der ASCII-Zeichensatz benutzt nur sieben Bits. Das achte Bit wird von verschiedenen Programmen für verschiedene Aufgabe eingesetzt. Der "Z"-Parameter ist nicht notwendig beim Kopieren nach ASCII-Einheiten, wie CON: oder LST:.

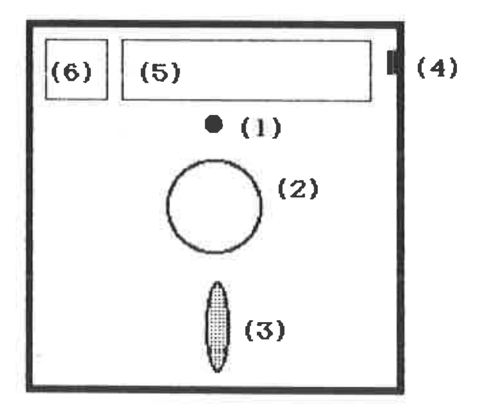
Anhang 4

Die SET-Parameter

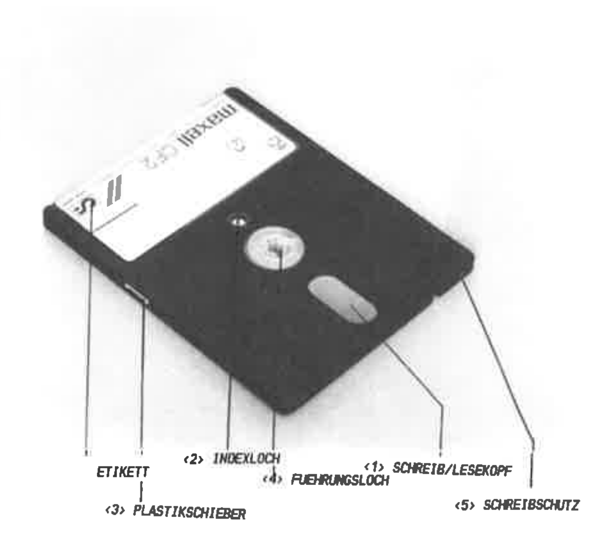
OPTION	BEDEUTUNG
DIR	Macht eine SYSTEM-Datei wieder im normalen Inhaltsverzeichnis sichtbar
SYS	Macht eine Datei zur SYSTEM-Datei
RO	Bestimmt, daß eine Datei nur gelesen werden kann
RW	Bestimmt, daß eine Datei gelesen und verändert werden kann
ARCHIV=OFF	Setzt das ARCHIV-Attribut auf "aus". Das bedeutet, daß diese Datei noch nicht gesichert (archiviert) wurde. Das Programm PIP mit der Option [A] kann Dateien mit mit dem Attribut ARCHIV=OFF kopieren. Den PIP-Befehl geben Sie mit den Sternchen für die Dateinamen ein und PIP kopiert alle Datei, die seit dem letzten Kopieren mit PIP und der ÄAÜ-Option verändert wurden. Nachdem PIP kopiert hat, setzt es die Datei-Attribute auf ARCHIV=ON.
ARCHIV=ON	Setzt das ARCHIV-Attribut auf "ein". Das bedeutet, daß diese Datei gesichert wurde. Normalerweise ändert PIP mit der Option [A] das Attribut nach dem Sichern von Dateien.Sie können das Attribut auch selber ändern, wenn Sie den SET-Befehl verwenden.
F1=ON/OFF	Schaltet das benutzerdefinierte Datei-

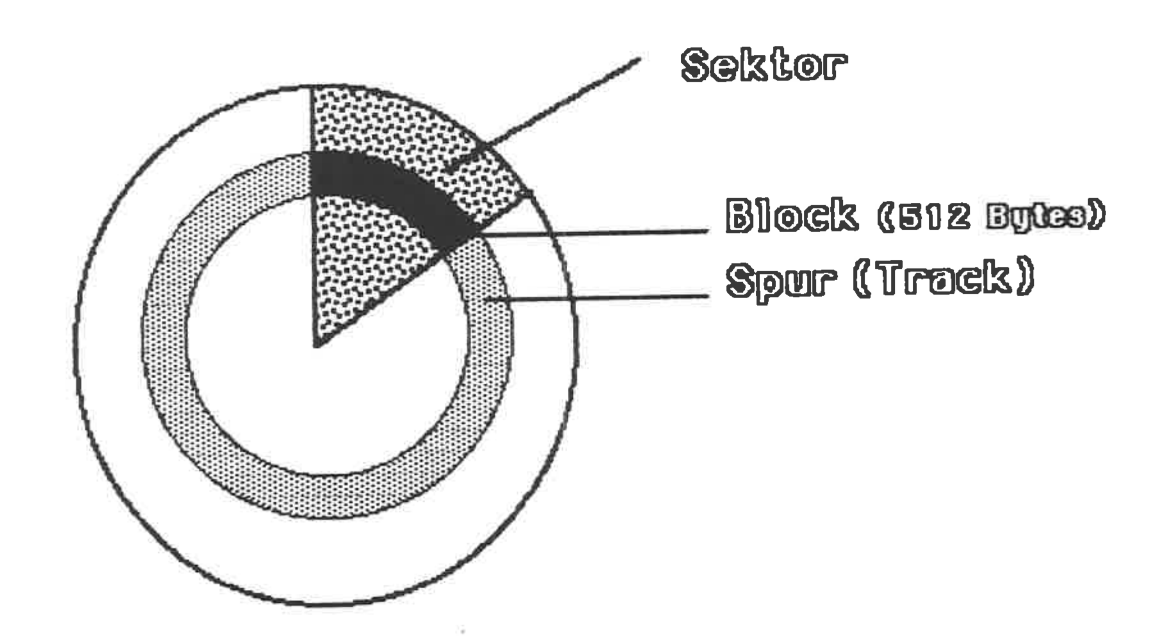
	Attribut F1 ein oder aus.
F2=ON/OFF	Schaltet das benutzerdefinierte Datei-Attribut F2 ein oder aus.
F3=ON/OFF	Schaltet das benutzerdefinierte Datei-Attribut F3 ein oder aus.
F4=ON/OFF	Schaltet das benutzerdefinierte Datei-Attribut F4 ein oder aus.

5-1/4 Zoll-Disketten



(1) Indexloch
(2) Führungsloch
(3) Loch für Schreib/Lesezugriff
(4) Schreibschutzkerbe
(5) Beschriftungsfeld (Temporär)
(6) Beschriftungsfeld (Dauerhaft)




Sektor
Block (512 Bytes)
Spur (Track)

DAS STEHT DRIN:

Endlich CP/M beherrschen! Von grundsätzlichen Erklärungen zur Speicherung von Zahlen, Schreibschutz oder ASCII über Anwendung von CP/M-Hilfsprogrammen bis zu “CP/M intern“ für Fortgeschrittene findet hier jeder CPC-Anwender schnell die notwendigen Hilfen und Informationen zur Arbeit mit CP/M. Dieses Buch berücksichtigt die Versionen CP/M 2.2 und CP/M PLUS (3.0) für Schneider CPC 464, CPC 664 und CPC 6128.

Aus dem Inhalt:

- Die Aufgabe von CP/M
- Die System-Diskette
- Regeln für Dateinamen
- Eingebaute Befehle
  USER, DIR, ERASE
- Transiente Befehle
  SET, PROTECT, SHOW, SUBMIT
- Alles über PIP
- Mehrere Dateien hintereinander drucken
- Fremde Diskettenformate lesen
- Unterschiede der CPC-Rechner

  und vieles mehr

UND GESCHRIEBEN HABEN DIESES BUCH:

Elmar A. Weiler ist freier Journalist, spezialisiert auf die leichtverständliche Darstellung komplizierter Materie. Nach seinen Büchern zu WordStar und dBase ist dies bereits sein drittes klar gegliedertes und ganz auf den Anwender zugeschnittenes Buch.
Jörg Schieb ist erfahrener Programmierer von Dateiprogrammen, Autor des Floppybuches zum CPC und kennt die Maschinenprogrammierung des CPC in- und auswendig.

ISBN 3-89011-089-4